

Design • Build • Program • Learn • Compete

SERVOMAGAZINE

FOR THE ROBOT INNOVATOR
www.servomagazine.com

MAGAZINE
July 2011

ROBOTS:
FORM OR
FUNCTION?
Which Is More Important?

◆ **MicroBot Surgeons**
Travel Inside The Human Body

◆ **The Robotist's Guide To Plastics**
What To Use And When

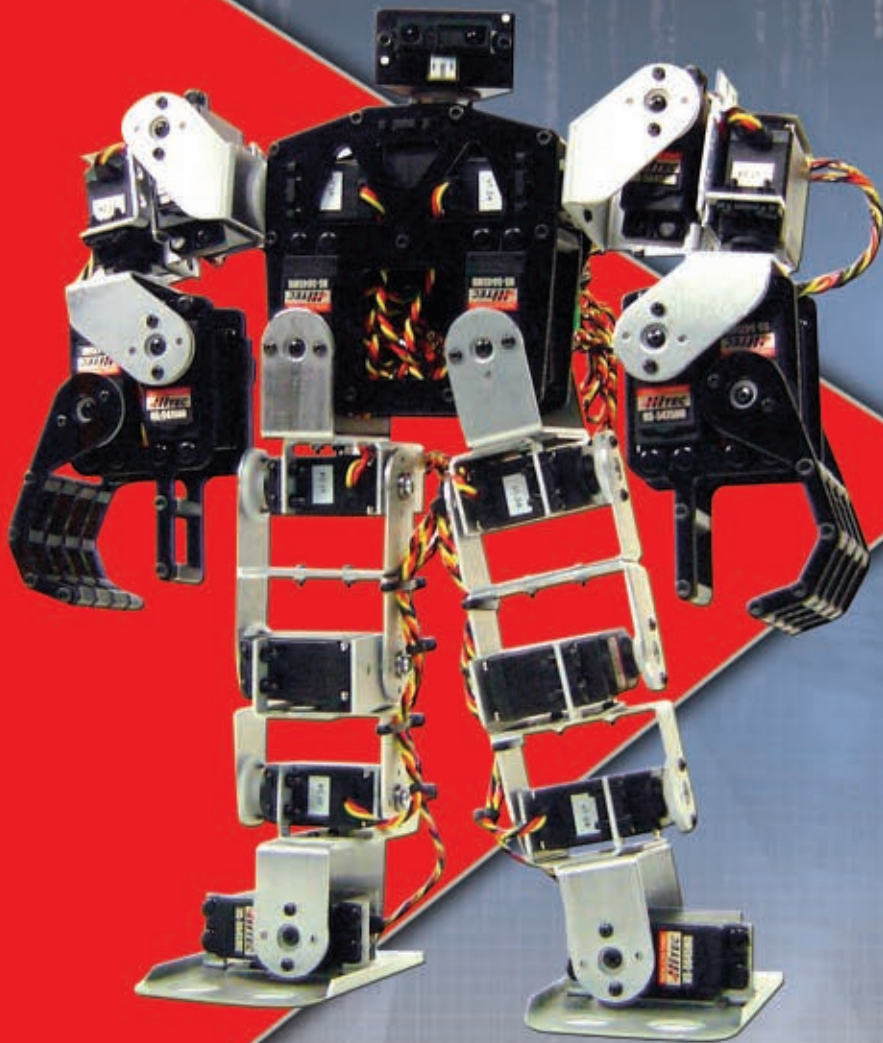
◆ **The NXT Big Thing**
Control Your NXT Robot Wirelessly By Waving Your Arms



\$5.99 US \$7.00 CAN

GET WITH THE PROGRAM!

Hitec's **HPP-21** and **HPP-21 Plus** are the perfect answer to programming your Hitec digital servos. Whether your robot uses our conventional **HS-5XXX/ HS-6XXX** servo series or the high voltage, ultra premium **HS-79XX** series, our programmers will dependably test and program all your digital servo parameters.



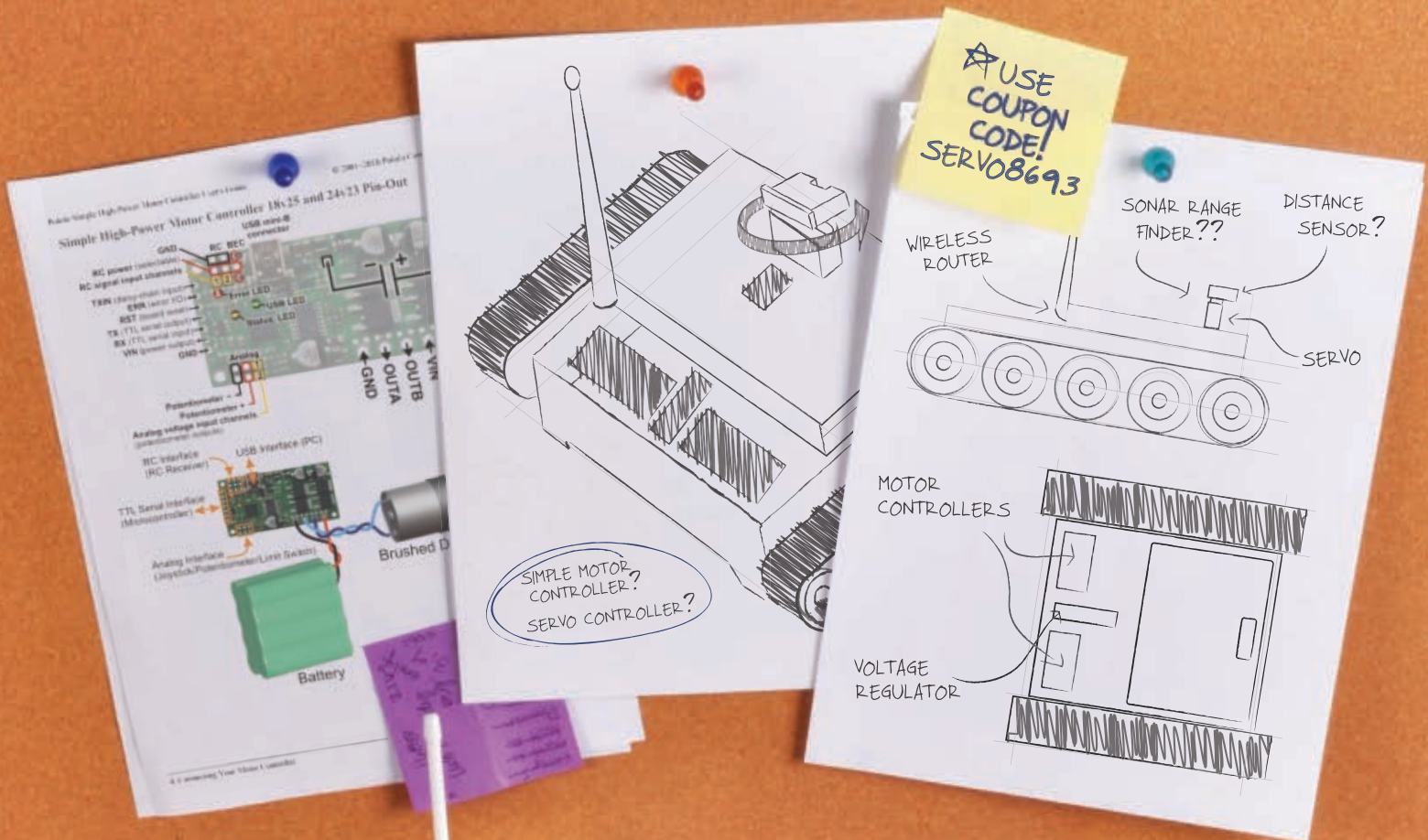
HPP-21
PC Programmer
for Hitec Digital Servos



HPP-21 Plus
PC and Field Programmer
for Hitec Digital Servos

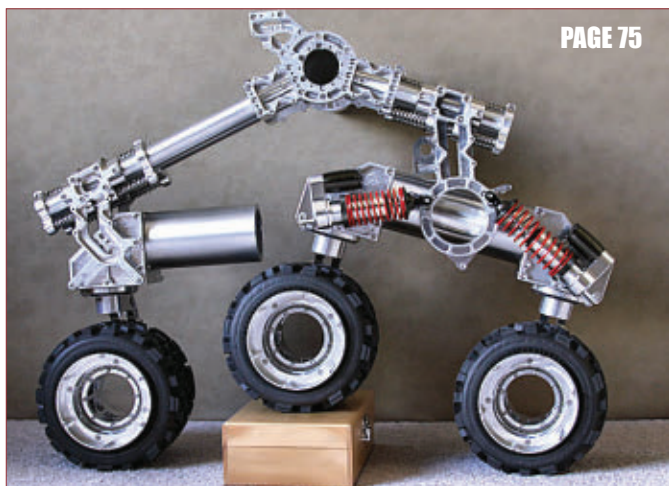
Both programmers will test servos of any brand and offer a USB-PC connection with free downloadable software.





Take your design from
idea to reality

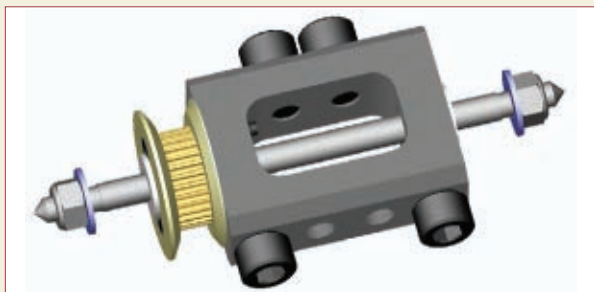
Did you know that each article in *SERVO Magazine* has its own webpage? It's where you go for downloads, comments, updates, corrections, or to link to the article in the digital issue. The unique link for each webpage is included with the article.



The Combat Zone...

Features

- 28 MANUFACTURING:**
The Robotist's Guide to Plastic
- 30 BUILD REPORT:**
Building a Better Beetle Beater Bar
— Part 1



Events

- 32 Upcoming Events**
- 32 EVENT REPORT:**
Robot Rumble

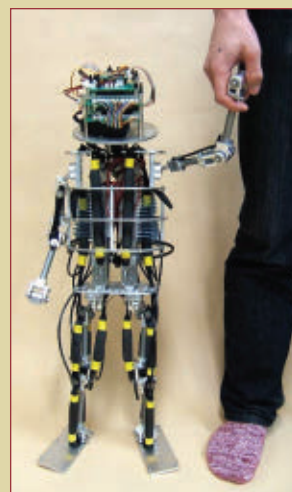
Columns

- 08 Robytes**
by Jeff Eckert
Stimulating Robot Tidbits
- 10 GeerHead**
by David Geer
*Miniature Surgeons Travel
Inside the Human Body*
- 14 Ask Mr. Roboto**
by Dennis Clark
Your Problems Solved Here
- 60 Twin Tweaks**
by Bryce and Evan Woolley
Savöx Rocks
- 70 The NXT Big Thing #9**
by Greg Intermaggio
Wii!
- 75 Then and Now**
by Tom Carroll
Robots: Form or Function?

Departments

PAGE 24

- 06 Mind/Iron**
- 20 Events Calendar**
- 22 New Products**
- 24 Bots in Brief**
- 66 Showcase**
- 67 SERVO Webstore**
- 81 Robo-Links**
- 81 Advertiser's Index**





PAGE 34

34 2011 VEX Robotics World Championship

by *R. Steven Rainwater*

Over 10,000 participants joined in this classic competition that included not only VEX events, but also BEST match-ups, Coast Guard AROW water robots, plus the first Boy Scouts to earn the new robotics badge.

38 Meet the Swiss Army Knife of Robotics: the Cerebot 32MX7

by *Fred Eady*

Whether it be robotics, remote control, telemetry, Internet services, CAN networking, or an I/O intensive application, the Digilent Cerebot 32MX7 board plugs physically and logically into the center of the action.

46 CPLDs — Part 5: A CPLD Project

by *David Ward*

For the last installment of this series, we incorporate a CPLD to control a mobile robotics platform.

49 Adding Smoke, Gas, and Heat Detection to Your Robot

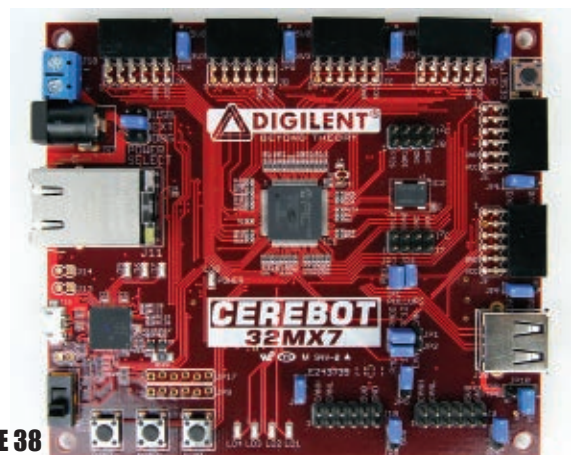
by *Gordon McComb*

Create your own "safety bot" by attaching sensors that can detect smoke, poisonous gasses, and high temperatures.

56 Adding Vision: Interfacing With RoboRealm

by *John Blankenship and Samuel Mishal*

For many hobbyists, giving their robot the gift of sight is one of their ultimate goals. RoboRealm is an application that can make adding vision painless by providing easy-to-use modules.



PAGE 38

Mind / Iron

by Bryan Bergeron, Editor



Voice Recognition Options

I've been having fun with SayIt: a voice recognition module available from Parallax (www.Parallax.com) for about \$60. The module is an affordable and painless means of adding voice recognition functionality to your robot or other microprocessor-based application.

SayIt supports 23 built-in commands which can be overridden by 32 custom commands of your choosing, and comes with an easy-to-use interface for the BASIC Stamp 2, as well as a training application. You can use the module as-is with other microprocessors, but you'll need a Stamp 2 if you want to add custom commands.

The voice recognition module brings back memories of my first voice recognition card from over 20 years ago. I forget the brand, but a card with functionality and vocabulary similar to that of SayIt was available for about \$60 for the PC. Although the card required most of the resources of the PC for operation, training it was simpler than training the SayIt because of a screen-based menu system. Today, the same functionality is available on a thumbnail sized chip running on an inexpensive microcontroller – with a total footprint less than that early voice recognition card.

Voice recognition isn't new, and robotics have become commonplace on the PC. For example, I'm dictating this editorial using the latest version of Dragon Dictate for Windows. If you're a Macintosh user, then you already have voice recognition software installed as part of your operating system. Unfortunately, the recognition accuracy of this 'free' software is less than stellar. You'd do better with an add-on product. In either case, dedicated hardware cards for speech recognition are a thing of the past, thanks to standard audio input/output hardware.

An advantage of using voice recognition to control your robot is that it frees your hands to do other things. Another advantage is that it enables those not versed in the nuances of their remote to operate their robot. It's one thing to have a robot that only responds to your voice, but you can get a lot of mileage out of a robot that will respond to the verbal commands of your friends and family, as well.

Voice recognition really shines when it's teamed with voice synthesis. A popular text-to-speech chip is Speakjet. If you look online, you can find numerous examples of interfacing the Speakjet to just about every microprocessor on the market. The output is distinctively 'robotic,' but in a robotics application, it's fitting.

The trick to working with speech as a user interface for both feedback and commands is to pick short commands that are both readily recognizable by the system and that can be readily learned by you and other users. For example, if you use simple one syllable commands (such as "right" and "go"), then recognition accuracy will be poor even though the commands are intuitive and easy to remember. If you use multiple syllable words that sound significantly different from each other, then recognition accuracy should increase considerably.

As a benchmark, 'good' accuracy in the current generation of inexpensive, discrete word, limited vocabulary voice recognition hardware is 90% or greater. That's one error in 10 words. The occasional error isn't a big deal if you're steering a carpet rover with voice commands. However, the error can become a serious matter for someone controlling a surgical robot or a weapon system. Even if you're just controlling a desktop robotic arm, it's a good idea to have a manual override for when the occasional misunderstanding occurs.

A more sophisticated workaround is to have the system repeat your command and wait for an acknowledgement. If you don't affirm the command within a few seconds, the system ignores the command. Depending on your robotic application and the complexity of commands, the tedium of acknowledging every command may be unbearable. I like the kill switch approach. It's immediate, and I can usually tell within a second or two whether my voice command has been properly interpreted.

The overhead imposed by speech recognition and – to a lesser extent – text-to-speech, can be significant on a robot tasked with monitoring multiple sensors and controlling several motors. A workaround is to dedicate a microcontroller to the speech interface. This frees your main microcontroller to deal with navigation and other primary tasks.

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX (951) 371-3052
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277, N. Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

**ASSOCIATE PUBLISHER/
VP OF SALES/MARKETING**
Robin Lemieux
display@servomagazine.com

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert
Tom Carroll
Dennis Clark
Fred Eady
Bryce Woolley
Gregory Intermaggio
David Ward
Samuel Mishal
Pete Smith

Jenn Eckert
David Geer
R. Steven Rainwater
Kevin Berry
Evan Woolley
Gordon McComb
John Blankenship
Matthew Spurrk

CIRCULATION DIRECTOR
Tracy Kerley
subscribe@servomagazine.com

**MARKETING COORDINATOR
WEBSTORE**
Brian Kirkpatrick
sales@servomagazine.com

WEB CONTENT
Michael Kaudze
website@servomagazine.com

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

PRODUCTION/GRAPHICS
Shannon Christensen

Copyright 2011 by
T & L Publications, Inc.
All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *SERVO Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *SERVO*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *SERVO*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: **430 Princeland Court, Corona, CA 92879.**

Printed in the USA on SFI & FSC stock.



If funds aren't a consideration, then the way to go is to handle the voice recognition and synthesis remotely on your PC. By sending the audio from your robot to a PC or laptop and having the audio processed there, you'll be able to use large vocabulary continuous speech to control your robot. Instead of single, low level commands, you can issue complex, high level commands such as "pick up the red ball." Of course, you have to have other resources (such as image recognition) to identify the red ball. You'll also need a macro language to translate "pick up" into a series of commands that are tied into feedback from sensors on your robot.

Obviously, developing a sophisticated human-robot interface based on verbal communications is no small undertaking. However, if you're new to this arena, experimenting with an inexpensive voice recognition chip is a great place to start. If you're using voice recognition and speech synthesis in your robotic project, please consider sharing your experiences with other readers. **SV**

THE ORIGINAL SINCE 1994

PCB-POOL®

Beta LAYOUT

Servicing your complete PCB prototype needs :

- Low Cost - High Quality PCB Prototypes
- Easy online Ordering
- Full DRC included
- Lead-times from 24 hrs
- Optional Chemical Tin finish no extra cost

NEW !

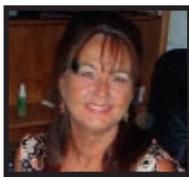
NEW !

FREE LASER STENCIL WITH ALL PROTOTYPE PCB ORDERS

Watch "ur" PCB®
Follow the production of your PCB in **REALTIME**

email : sales@pcb-pool.com
Toll Free USA : 1 877 390 8541
www.pcb-pool.com

Beta
LAYOUT



Robytes

by Jeff and Jenn Eckert

Scouting for Robots

A lot has changed within the Boy Scouts of America (www.scouting.org) since it was founded in 1910, and it's no longer just a bunch of kids tying knots and rubbing sticks together. These days, a scout can earn merit badges in such things as nuclear science, oceanography, composite materials, cinematography, and entrepreneurship. (For the less technically ambitious, choices also include dog care, disabilities awareness, basketry, and salesmanship, but never mind.) The latest introduction to the list is robotics, the requirements for which were developed in



No, this doesn't depict a mutant raccoon crawling out of a wheat field. It's the Boy Scouts' new robotics merit badge, ostensibly resembling NASA's Mars Rover.

consultation with organizations that include NASA, iRobot, Carnegie Mellon, and LEGO. To earn the badge, one must — among other things — discuss three of the five major robotics fields (human-robot interface, mobility, manipulation, programming, sensors), explain how robots are used today, and (of course) design, build, program, and test one. The BSA figures it will take only about 14 hours to meet the requirements, so we're not talking about ASIMO. It is such a popular subject, however, that the organization expects to issue more than 10,000 badges during the first year.

Sort of a Robot

It's not clear how you start with the contemplative enlightenment of Zen Buddhism and end up in Helsinki sorting trash, but the folks at Finland's ZenRobotics (www.zenrobotics.com) have made the journey. According to the company, "Our world continues to drown in trash. What's more, many of the raw materials we rely on will run out in our lifetime should we fail to establish a solid recycling process. But fear not! The robot boffins at the explosively growing ZenRobotics Ltd. are going to save the world with intelligent robots." The ZenRobotics Recycler is a conveyor system that uses artificial intelligence and a variety of visual sensors, weight measurements, spectrometry, metal detection, and a robotic arm with tactile feedback to separate junk from usable materials in construction and demolition remnants. It recognizes and sorts a range of materials including plastics, concrete, wood, and metal. Semi-interestingly, the unit was originally dubbed the Zendroid, but some legal toady deep in the bowels of Google thought it sounded too much like "Android" and fretted that the public would get the two confused. A lawsuit was threatened and — with zen-like tolerance — the Finns let the name drift into oblivion. Their website is littered with little bits of dry humor, such as the ever-changing range of aphorisms floating under the company logo, such as "Inventing Slogans Since 2007." To see their droll video trailer, go to www.youtube.com/watch?v=bPDH2uSaQeY.



The ZenRobotics team shows off a gripper.



The three color modes of the DON-8r bot.

Moocherbot Hits the Streets

A 21 year old student at the University of Dundee, Scotland, Tim Pryde (www.timpryde.com), has pushed the boundaries of robotics into the realm of street begging. With Orwellian irony, Tim named his creation DON-8r (as in "donator"), although BUM-r might be more accurate. First, the little beggar displays a green light and moves randomly for a set length of time, after which it stops, waves a flag, flashes red, and repeats "hello." This continues relentlessly until someone slips a coin into a slot in his back, after which he thanks the mark and flashes blue. Then, the routine starts over. Tim intends to rebrand the units to suit individual charities, so you might eventually see them sporting Red Cross emblems or Salvation Army uniforms. There appear to be a couple bugs to work out, though. For one thing, DON-8r is said to be equipped with collision avoidance capabilities,

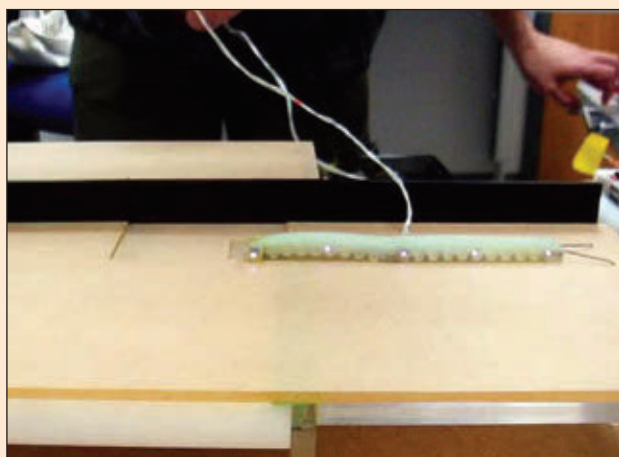
but the video on the website ends with him wandering into traffic, requiring human intervention. Another potential problem is that the bot is lightweight and defenseless, so it wouldn't be too difficult for a competing human panhandler to crack him open like a piggy bank and make off with the loot. But maybe no one else will think of that.

FUE on You

If you happen to be follically challenged and looking to be refuzzed, there may be some good news. In days of yore, the hair transplant procedure involved removing a strip of hair from the back of your head, chopping it into smaller chunks, and planting the chunks into barren regions. This doesn't look particularly natural, plus it leaves a scar across the back of your bean. A newer procedure — called follicular unit extraction (FUE) — involves harvesting smaller sections (usually containing only two to six hairs) for relocation. Unfortunately, it takes quite a bit of time for the doc to slice out the thousand or so units he needs. However, the US Food and Drug Administration recently approved the ARTAS System from Now Restoration Robotics (www.restorationrobotics.com) to assist with the job. The device is described by the manufacturer as "an interactive, computer-assisted system using image-guided robotics to enhance the quality of hair follicle harvesting." Its robotic arm punches out pieces of your scalp faster than a human can do it and with less damage (transection) to healthy follicles, so its cost is expected to be offset by the reduction in required time and personnel. The downside is that it doesn't actually do the transplant part of the procedure, so the doctor will still have to sow what he reaps manually. Because ARTAS was tested only on people with black or brown straight hair, it's not approved for use on anyone else.



The ARTAS robotic hair harvesting system.



GoQBot, inspired by the Crambidae caterpillar.

Robot Goes Ballistic

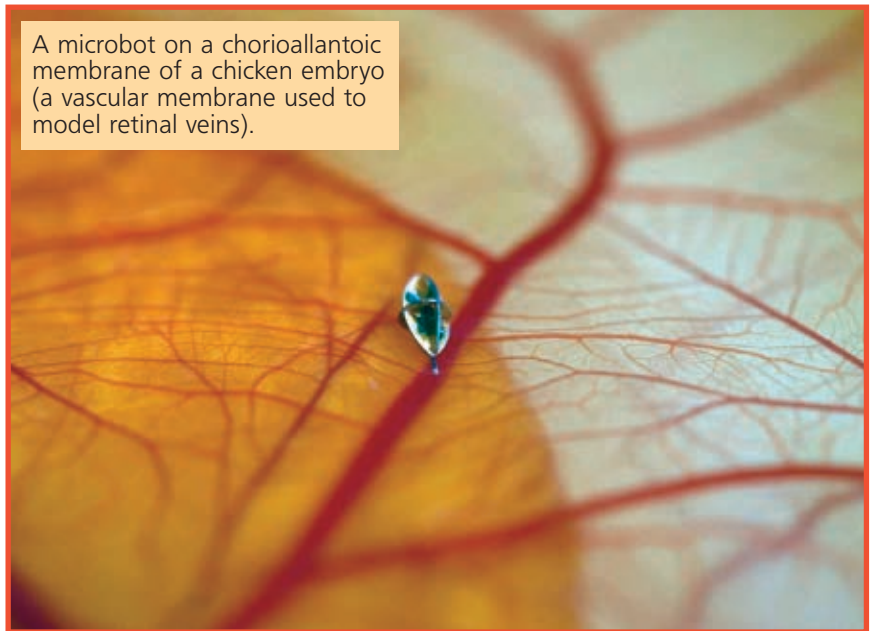
A few of my distant relatives still live deep in hill country, eat squirrel brains for breakfast, and are deeply fearful of hoop snakes — serpents that, according to legend, bite their own tails so as to form a hoop, roll after their victims like a wheel, and at the last second unclasp their tails so as to strike. Although no one this side of a moonshine jug has ever seen one, it turns out that some caterpillars really can move in a similar manner. Using a process known as "ballistic rolling," they can curl into a wheel and propel themselves away from predators at a speed that far exceeds their top crawl rate. As we know, whenever roboticists witness any kind of strange behavior in nature, they are driven to create a mechanical version of it, and so some researchers at Tufts University (www.tufts.edu) have devised a 10 cm long, soft-bodied silicone rubber robot called "GoQBot." (The name comes from the fact that it forms a "Q" shape before it begins to roll.) Actuated by embedded shape memory alloy coils, GoQBot is capable of squishing along at over 0.5 m/s (19.7 in/s). Such a device might seem to have little practical application, but according to Huai-Ti Lin, of the Tufts Department of Biology, it has potential uses in search-and-rescue operations. "Due to the increased speed and range, limbless crawling robots with ballistic rolling capability could be deployed more generally at a disaster site such as a Tsunami aftermath. The robot can wheel to a debris field and wiggle into the danger for us." You can take a look at www.youtube.com/watch?v=wZe9qWi-LUo. **SV**



Miniature Surgeons Travel Inside the Human Body

Microbots — wireless miniature surgical robots — traverse veins and ventricles performing life-saving procedures inside the human body. Atop a foundation of benefits of minimally invasive surgery, scientists are sampling the efficacy of sending mechanical physicians of only a few millimeters into orifices, organs, and blood vessels to administer treatments with little or no disruption.

A microbot on a chorioallantoic membrane of a chicken embryo (a vascular membrane used to model retinal veins).



Current mini-medical robots perform capsule colonoscopies and work in the GI tract. By shrinking the robots to a few millimeters, researchers and clinicians can start to think about working inside the urinary system, the circulatory system, and even the central nervous system, as well as blood vessels in the eyeball (*Microbots for Minimally Invasive Medicine*, Nelson, et. al., April 2010, AnnualReviews.org).

Procedures include therapeutic drug delivery, highly localized radiation therapy for tumors, delivering stem cells, opening blood vessels, taking minute biopsies, and inserting scaffoldings, stints, and implants to name a subset of theorized and as yet unknown possibilities; some are already successful in animal trials.

Using wireless technology, the medical robots would transmit areas of interest such as cancers discovered using sensors, as well as the bot's location in the given human system. With the bot's location at the time of discovery, physicians could pinpoint the cancers, internal bleeding, and other matters of concern. The robots are highly applicable to microsurgery of the retina, which is currently difficult to

approach. The robots could use the low level force and reach the heights of precision necessary, neither of which is possible with the human hand. A robot is theorized that could be observed through the eye and manipulated using magnetic wireless technology.

Microbots could also perform operations on the inner ear, where the bots could install cochlear implants with increases in sound quality. Similar procedures using current surgical technologies can lead to damage and infection.

Microbots could be used in a number of fetal surgeries where existing surgical tools lose their degrees of freedom due to the number of layers (mother's abdomen, uterus) the tools must pass through.

Power Up and Go

Potential power supplies for microbots include onboard power such as thin film batteries, MEMS-based generators, and "scavenged" power in which the robot pulls energy from chemicals in the environment, i.e., inside the human body where the pH is neutral (cited in Nelson, et. al.).

Power could also be transmitted to the robots wirelessly using magnetic fields. Methods include time-varied magnetic fields that generate electricity, and quasi-static and low frequency magnetic fields that apply a torque to a magnetic material. In either case, magnetic fields would pass through the human body as they do through air vacuums.

Artificial bacterial flagella that imitate real flagella that microorganisms use to swim in the human body would best provide locomotion. IRIS has developed a helical swimming robot that consists of a helical tail and a soft magnetic metal head.

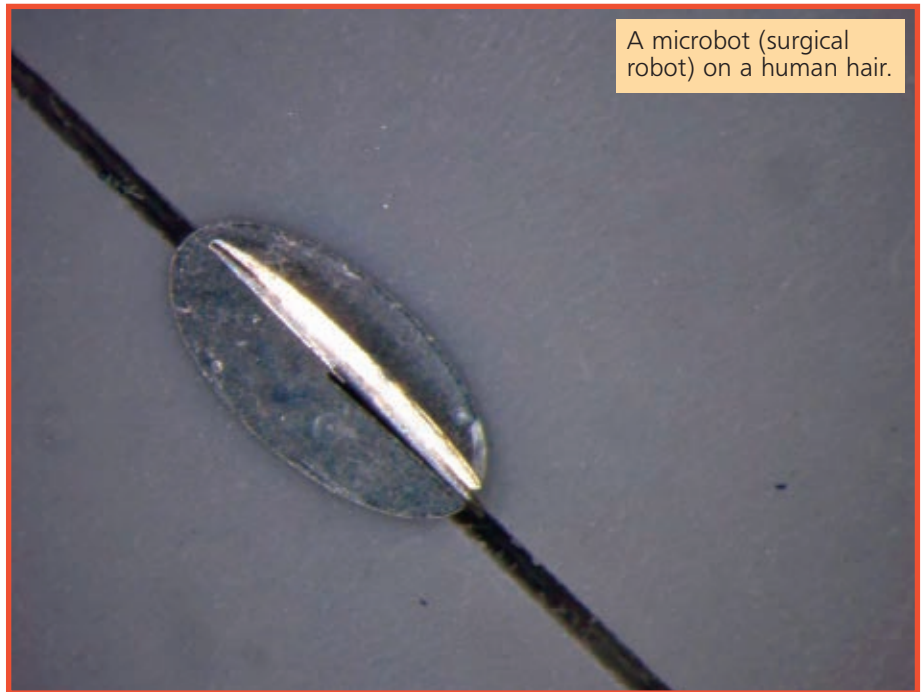
Specifically, magnetotactic bacteria or MTB could be harnessed by MRI systems to achieve robot locomotion by turning the microbot in a screw fashion using a rotating magnetic field in an MRI machine. Physicians could adjust the rotation speed and direction of the magnetic field in order to change the velocity and direction of the robot while in a human system. The clinician could invert the magnetic field rotation in order to reverse the direction of the microbot.

Scientists propose to harness the MTB by attaching nanoparticles, adding a payload inside the MTB, using individual, attached MTB to push or pull the robot, or using a swarm of unattached MTB to move the microbot.

A number of microswimmer designs for microbot locomotion are based on helical propellers. The design employs a rotary actuator to turn a propeller to move the microbot. One design uses a rotary motor to turn a "passive flagellum." Researchers have shown that a rotating magnetic field can control a magnetic helical propeller.

The propeller can be fabricated in a few different ways. Researchers can make it rigid, by forming a wire into a helix. They can also use a multilayer strip that forms itself into a helix due to internal stresses. In both cases, the propellers can push or pull the microbot by changing the direction of the propeller's circular motion.

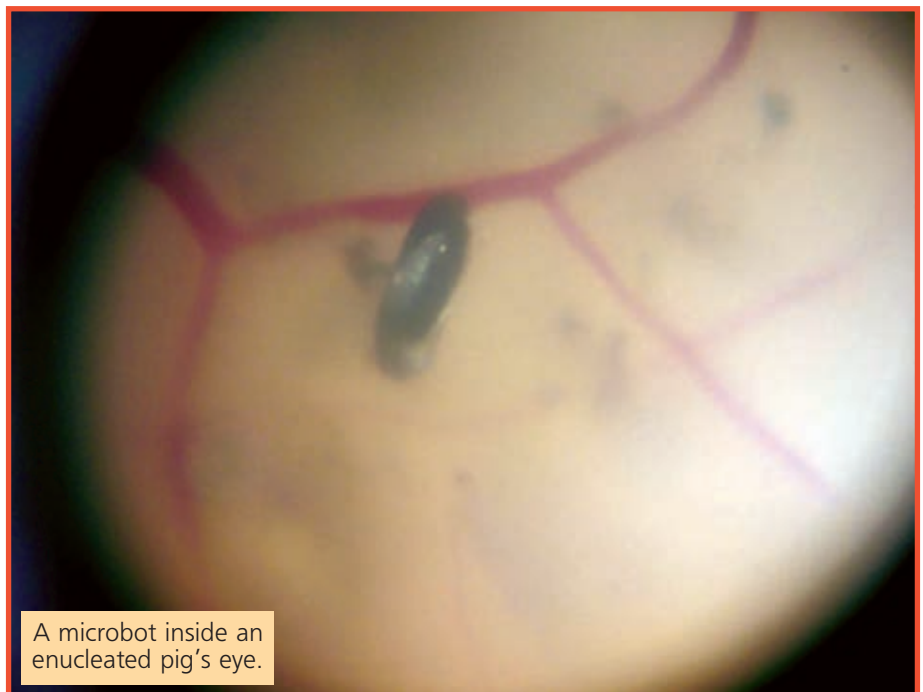
The propellers can be constructed to be compliant with a helical geometry only in a dynamic fashion. A fiber under a certain amount of torsion will form a helix which will remain until the force creating the torsion is removed.



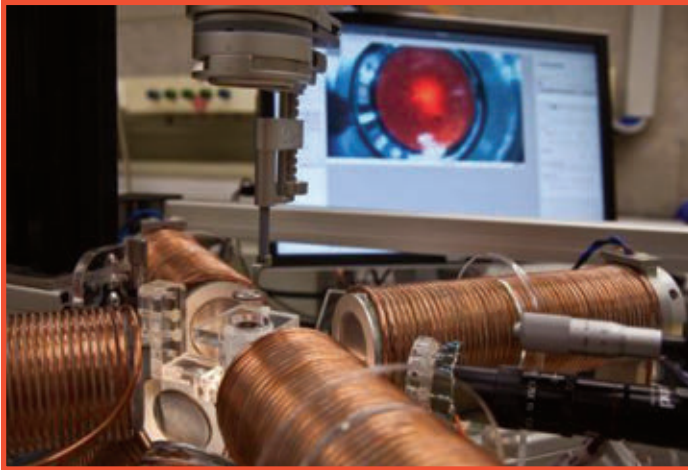
A microbot (surgical robot) on a human hair.

Propulsion using this type of helix is less controllable and is only easily applied to pushing the microbot, not to pulling it. However, this type of helix has advantages. For example, this approach permits researchers to use long strands of a drug that twist into a helical shape such that the payload of the microbot (the drug delivered) becomes the propeller.

Helix propulsion systems can also wrap around the microbot such that the bot can crawl through a human system in a circular motion like a bolt screwing into a nut.



A microbot inside an enucleated pig's eye.



This is an image of the Octomag. In the background, you can see an image as viewed through an ophthalmic microscope.

Distinct from this approach is a screw shaped microbot with embedded magnetic material that physicians could some day drive through tissue wirelessly. This approach can enable moving through soft tissue and not just inside veins. This approach requires the helix to make only one full rotation to move the length of the microbot through human tissue. The other helix-propelled bots must turn the helix several times to move through human fluid.

Another propulsion method for microbots is the traveling wave. Traveling wave propulsion looks like a fish tail swimming behind the microbot. Traveling waves are more efficient at propulsion than propellers, assuming the efficiency of the actuation is equal to that of the propeller system. However, there are fabrication, power, and control issues related to creating distributed actuation for the traveling wave method. Methods that create traveling waves without distributed actuation are more feasible but less efficient.

Modular, Self-Assembling Mini Medical Robots

In another approach to microbots, researchers are adding active locomotion and telemetry to capsule endoscopy technology. According to IRIS, the size of the capsule cannot be larger than what a person can swallow. In this new method, the larger robot would be swallowed in individual pieces that would assemble themselves inside the gastrointestinal tract. Self-assembly is the biggest challenge to this approach.

By using magnets on the faces of the modules that

must connect to one another, researchers hope to solve the self-assembly problem. The results are promising for a snaking robot that can assemble itself and make its way through the small intestine.

The self-assembling robot would have sensors on its front end to detect pathologies that would wirelessly communicate to the surgeon or physician. A module designed for specific tasks would execute commands to accomplish the task once an actuation module moves the robot into the right place. Sensors and actuators that work on individual pill sized devices are already being developed. Each sensor or actuator could eventually constitute a module in the larger robot discussed here.

Eye See the Robot

One application of microbots is retinal therapies where the robot could help the surgeon operate inside the human eye. The primary challenge to this approach is localization. Because researchers and clinicians can easily see what is in the eye, they can look through the eyeball to localize the robot. Using high quality imaging of the rear of the human eye, they can develop algorithms to track and localize the microbots. However, there is no existing research that explains what the effects of this will be.

IRIS researchers have developed a mechatronic vitreoretinal ophthalmoscope that can take an image of the entire retina despite the obstruction of view created by the other parts of the eye. They can use this device to image and follow objects in the retina. There is on-going research to discover how much illumination will be necessary to follow objects in the human eye.

The imaging device must be able to differentiate the microbot from the rest of the retina. The mechatronic device must use several lenses to see beyond the other components of the eye to keep the retina in view. Images are still blurry much of the time. The light that reaches the eye's interior can decrease the quality of the images taken due to uneven brightness. The microbot has no distinctive colors to distinguish it from the rest of the eye.

IRIS researchers have surmounted many of these obstacles by using trained color histograms, colorspace evaluation, and maximum separability thresholding. According to IRIS, a real time color tracker in the right color space assures that the microbot can be separated from the rest of the eye.

Conclusion

Production-quality microbots are on the horizon. Soon, a variety of surgeries will be possible using these magnificent miniaturized microbots. **SV**

Resources

Institute of Robotics and Intelligent Systems
www.iris.ethz.ch

Research of Professor Bradley J. Nelson
www.iris.ethz.ch/msrl/research

DF RobotShop Rover

Compatible with



Affordable, Flexible and Scalable

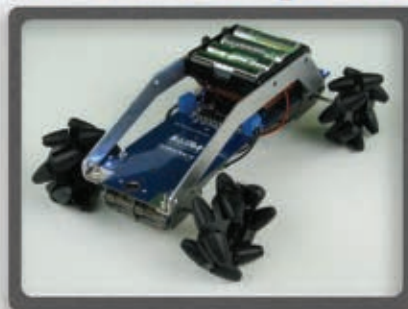
Compatible with thousands of off-the-shelf parts!



Wireless Control



Speech Control*



Mecanum



Scan on your smart-phone
with a bar reader app

**ADDITIONAL VARIANTS TO COME
THE EVOLUTION CONTINUES...**



Our resident expert on all things robotic is merely an email away.
roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

ASK MR. ROBOTO

by
Dennis Clark

Robot competition season is upon us. Just check the *SERVO Magazine* calendar to find the one closest to you. As I write this column, the SparkFun AVC (Autonomous Vehicle Competition) has completed and the results have been posted. This competition is a bit different than most you see. The rules are simple, and occasionally capricious folks come from all over the US, and some from other countries as well to compete. All your robot needs to do is circumnavigate the SparkFun building in the fastest time to win.

There are two categories: ground and aircraft. In this third year of the competition, SparkFun estimates that nearly 700 people showed up to compete or watch robots. I don't know the full list of competitors, but the ground crowd was somewhere around 40 or more robots. The aerial class had about 10 there, but not all actually got into the air. Since this is in Colorado, you can't trust the weather ... ever. On the day of the competition, it was cold and there was a light snowfall. As the day went on, we moved to rain and in the afternoon we actually saw some daylight. Needless to say, robots had to be able to deal with moisture in the air and on the ground.

It seemed that the clouds hampered folks' GPS units since many vehicles had problems getting position locks. Only two aircraft finished the course; many crashed and some just wandered off into the wild blue yonder (and amazingly survived!). We had some very entertaining entries (like AV Saurus Rex and Dr. Zeus' Magical Flaming Banana). The SparkFun band was playing and everyone enjoyed themselves (when they weren't freezing). There was a lot to see and I don't remember the names of all of the robots and their teams, but I've included some shots of the event here.

In short, the winner of the ground competition was Team TOBOR (puzzle time ... what is that backwards?). In the aerial division, Team Robota finally came out on top. This is their second year in a row to win. Team Robota is clearly the team to beat in the air. SparkFun gave a few more awards for those robots that entertained the crowds, and they were:

- Crowd Pleaser (Most Entertaining): AV Saurus Rex.
- Engineers Choice (Best Technical Design): Plan B.

- Kill Switch Award (Most Dangerous): DIY Drones.
- Water Hazard Award (Finds the Pond and/or Stream): Jombies.
- Rookie Award (Best First-Time Competitor): Viator.
- Best Dressed (Most Interesting Costume/Vehicle): Dr. Zeus' Magical Flaming Banana.

SparkFun also gave tours, so we got a glimpse into how innovative the place really is. SparkFun is filled with interesting rooms where interesting projects become products, and where products get turned into interesting projects, as well. I recommend the tour for newcomers. It shows how a small company can respond to the needs of hobbyists' whims and become a profitable business. (Take THAT huge megacorps!) You can see in the photos how fascinated the next generation of engineers can be with their stuff. And just when you think you've seen it all, SparkFun comes out with a way to help MIT stock cool stuff for student projects: a vending machine. What a cool idea!

The weather at AVC 2011 tried its best to dampen enthusiasm, but robots STILL won the day. Food vendors, lots of pavilions to protect robots (and their creators) from the elements, and bleachers (which no one used, you know, kind of wet ...) helped everyone keep the human's spirits up (no one asked the robots how they felt about the wet.) At the end of the day, an open invitation





One of the quad-copter entries.



Magical Flaming Banana.



Team TOBOR.



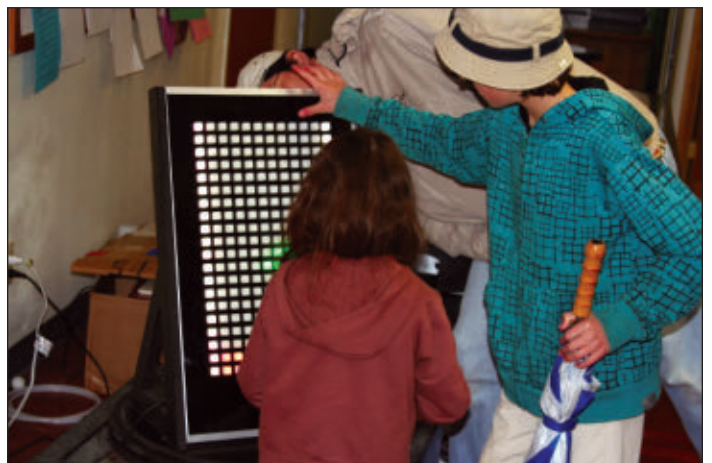
AV Saurus Rex about to crash.
The crowd isn't exactly running in fear.



Team Robota (I think ...).



The smallest entry.



Kids at heart create cool things for, well, kids!

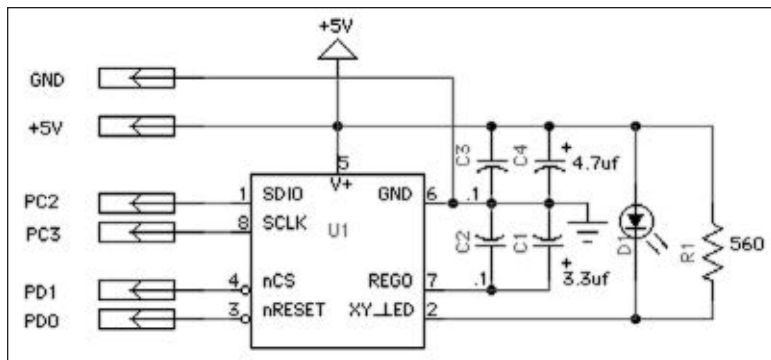


Figure 1. ADNS-5050 circuit.

from the SparkFun crowd to join them at a local pub for celebrations and spiritual augmentation sent everyone off with a smile and fond memories of another notable AVC event.

Mr. Roboto continues his answer from the May and June columns

Let's move on with our usual Mr. Roboto! As you may remember, I completely failed to successfully hack a

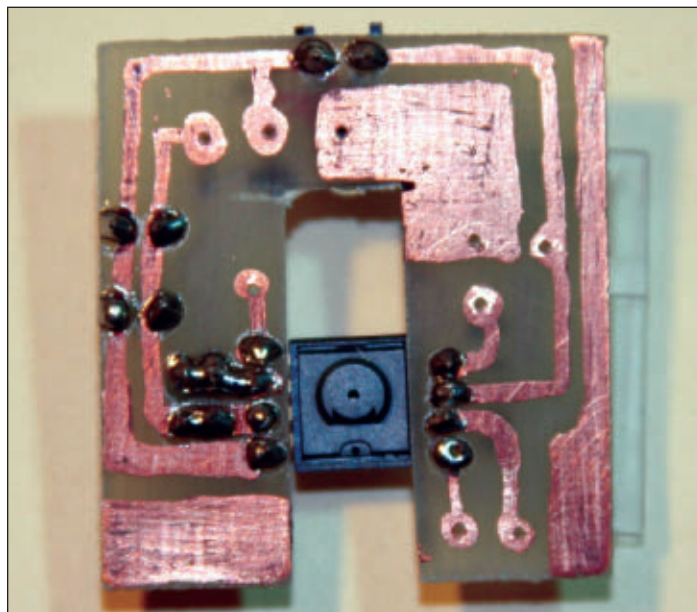


Figure 2. DIY optical sensor board.

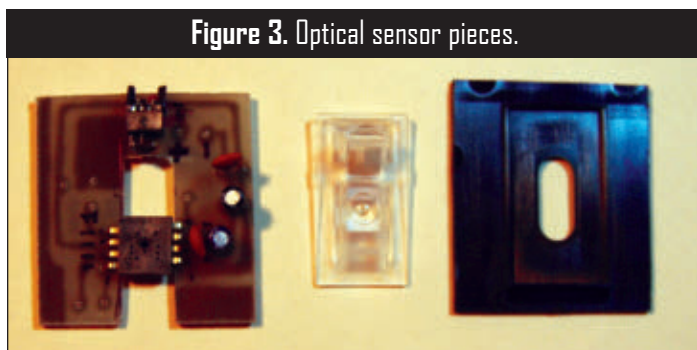


Figure 3. Optical sensor pieces.

PARTS LIST

Part #	Digi-Key	Cost
Avago ADNS-5050 Optical Sensor	516-2261	\$1.60
Avago ADNS-5100-001 Trim Lens	516-2300	\$0.57
Avago HLMP-ED31-SV00 Red LED	516-1372	\$0.77
Avago HLMP-5029 LED Clip	516-1395	\$3.78 (for 10)

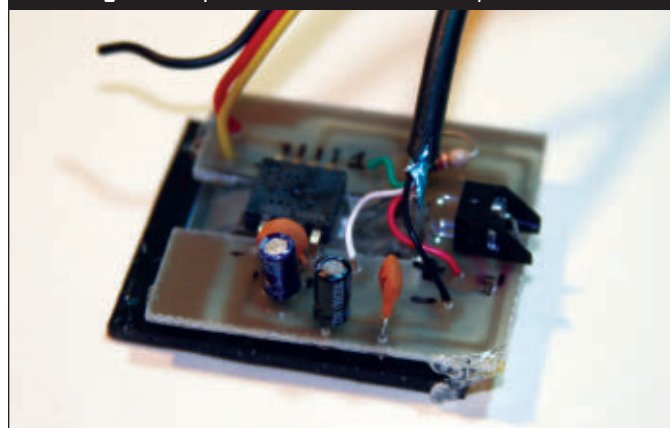
commercial mouse into a workable motion tracking device. Shameful, it was. However, not to give up I decided that instead of hacking stuff out of a mouse, I can just *make* a mouse from parts available at Digi-Key. I'm here to say that THAT project went swimmingly, and I've got some code and design

information to help you do it too! The **Parts List** shows a re-cap of what I'm going to be using.

Obviously, this is a bit cheaper than buying a mouse and then tearing it apart. As a bonus, the ADNS-5050 can be used as a very low resolution camera. It can take and send an 18x18 pixel grayscale image via the SPI port. This might be useful for motion detection or line following, and even horizon detection – you would just need a different lens to focus on something further away. (This bears consideration!) At this time, I was interested in some kind of dead-reckoning odometry. In **Figure 1**, you can see how simple the wiring of the ADNS-5050 sensor is. I built mine on a scrap piece of single-sided copper-clad board that you can get at a variety of places. There wasn't much to go on the board, so I just picked up a common Sharpie pen and drew a quick (and pretty dirty) circuit on the board and etched it on the bench. The results of that board with the parts installed is shown in **Figure 2**.

As you can see back in **Figure 1**, the circuit design isn't difficult, and the parts layout isn't critical either. I used the reference design given in the datasheet for the ADNS-5050. (Digi-Key has it online when you look up the part.) The ADNS-5050 datasheet calls for D1 to be the HLMP-ED80, but Digi-Key didn't stock it. I chose the HLMP-ED31 as a substitution based upon its characteristics, and it works fine. The reference design calls for R1 as "optional," and I don't know what it is there for, but it works and I noticed that the two other mice I

Figure 4. Optical sensor final assembly with wires.



eviscerated also had a resistor in parallel with their LED so this must be a common design option.

If you read the ADNS-5050 datasheet carefully, you'll find that the device has an "idle" current of 9-11 mA and the XY_LED pulls 45 mA to 51.5 mA (ouch!) which can drain a battery pretty fast. So, make sure you are using one with the capacity to deal with this sensor for the duration of your robot's running. I have not verified the average current draw, so it could be that the sensor can "throttle back" the current when nothing appears to be happening. You have been warned!

After building the printed circuit board (PCB), I needed to mount it and properly align the lens with the sensor, as well as space it correctly above the surface it will be dragging across. **Figure 3** shows my final solution. I cut the bottom out of one of my aborted attempts at a mouse hack which had a nice indentation that matched the size of my lens assembly. I then attached the PCB to the bottom plate with my favorite robot adhesive — a hot-glue gun (**Figure 4**).

Okay, I admit, this isn't pretty. I could have done a better job with the wiring layout to make the cabling better, but, Mr. Roboto has a time restriction on his projects so they can be published in a timely manner, so some allowances are occasionally made in the name of expediency. The assembly is easy to maneuver around to check how well it works, too.

The ADNS uses a form of SPI (serial peripheral

```
void InitADNS(void)
/*
 * Sets up the controller and does a reset on the ADNS-5050
 */
{
    DDRC |= 0x0C;
    DDRD |= 0x03;
    // Reset the chip
    nADNSRS = 0;
    _delay_us(2);
    nADNSRS = 1;
}

char ADNS_wr(unsigned char addr, unsigned char data)
/*
 * ADNS5050 write
 * addr = register address
 * data = data to write there or nothing
 * returns data read back (or nothing if a write)
 * msb of address = 1 for a write, 0 for a read
 *
 * We are operating at about 500K bps clock rate, it's about 20us to
 * do a full read/write session. This can obviously be improved upon.
 * The ADNS-5050 reads/writes the msb first, we work from the "left"
 * side to the "right" when writing and the reverse when reading back.
 */
{
    char temp;
    int n;

    nADNSCS = 0; // select the chip

    temp = addr;
    SCK = 0; // start clock low
    DATA_OUT; // set data line for output
    for (n=0; n<8; n++) {
        SCK = 0;
        _delay_us(1);
        if (temp & 0x80)
            SDOUT = 1;
        else
            SDOUT = 0;
        temp = (temp << 1);
        SCK = 1;
        _delay_us(1); // short clock pulse
    }
    if (addr & 0x80) { // This is a write, keep writing
        temp = data;
        for (n=0; n<8; n++) {
            SCK = 0;
            _delay_us(1);
            if (temp & 0x80)
                SDOUT = 1;
            else
                SDOUT = 0;
            temp = (temp << 1);
            SCK = 1;
            _delay_us(1); // short clock pulse
        }
    }
    else { // This is a read, switch to input
        DATA_IN;
        for (n=0; n<8; n++) { // read back the data
            SCK = 0;
            _delay_us(1);
            SCK = 1;
            if (SDIN) { // got a '1'
                temp|=1;
            }
            temp = (temp <<1); // shift left
            _delay_us(1);
        }
    }
    nADNSCS = 1; // de-select the chip
    return(temp);
}
```

LISTING 1. ADNS-5050 setup and control code.


```

/*
 * register addresses
 */
#define PROD_ID      0x00
#define REV_ID       0x01
#define MOTION       0x02
#define DELTA_X      0x03
#define DELTA_Y      0x04
#define SQUAL        0x05
#define P_GRABR      0x0B
#define P_GRABW      0x8B
#define M_CTRLW      0x8D
#define M_CTRLR      0x0D
#define M_CTRL2W     0x99
#define M_CTRL2R     0x19
#define LED_DC        0xA2
#define BURST        0x63
#define INV_ID       0x3F

/*
 * Data values
 */
#define ID5050        0x12          // product ID for ADNS5050
#define GOTMOT        0x80          // msb set, motion since last read
#define C_PDN         0x02          // (or) to power down
#define C_RES1K       0x01          // (or) for 1000 cpi
#define DC_MODE       0x80          // LED always on

```

LISTING 2. Select ADNS-5050 control registers.

interface) to communicate. It appears that it will handle about a 5 MHz clock rate, with various timing restrictions to account for response times. I used a 500 kHz clock rate because it was easy to implement using a bit-banged interface. I am using an Atmel ATMEGA168 part and didn't want to worry about interference with the SPI programming interface when programming the microcontroller; that would require more care to be taken to make sure that the ADNS-5050 didn't respond to the programmer commands. **Listing 1** shows how I set up and talked to the ADNS using my bit-banged command.

Because I wanted a self-contained demo unit, I decided upon the Pololu Orangutan because it has a built-in 8 x 2 LCD display. Pololu supplies a library with the board that

provides convenient calls to operate the on-board circuitry, like the LCD.

The ADNS-5050 is controlled by writing to and reading from selected registers. **Listing 2** shows the registers that I thought would be useful for this project — I didn't use many of them, but I did experiment with them a bit.

The fast explanation about how to use this device is to first read the MOTION register (0x02) and check bit 7. If it is set, then there has been a change of location and you then read the DELTA_X and DELTA_Y registers (0x03 and 0x04). However, I never saw bit 7 set when I read this register — but I would then read the DELTA_X and DELTA_Y registers and if they were non-

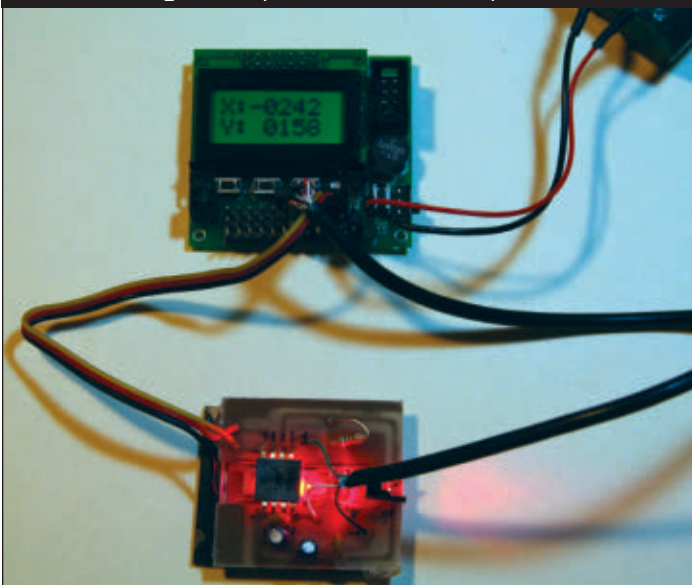
zero, then something obviously happened. Regardless, you MUST read 0x02 before anything will be in the DELTA_X and DELTA_Y registers. I left the device in its default setting (500 cpi) which I'm assuming is 500 counts per inch; that is what it looked like when I experimented with my assembly.

Listing 3 shows my main code. I continually accumulated the values read from the ADNS-5050 (which are signed eight-bit values) as I sampled it every 10 ms. My readings were reasonably repeatable using speeds commonly attained by table-top robots.

I must mention though that repeatability depends upon how "interesting" your surface is. I found that my results varied significantly with surface texture and grain. When I tested on a surface with a uniform texture (a manila folder for instance), the sensor was pretty repeatable. When I tested on a simulated wood-grain Formica surface, however, the results were not terribly repeatable. So, if you want to use this type of sensor for odometry, you need to control your surface or your results will not be good. It is possible that you could put a disk on your wheels of a regular texture and mount the sensor to read that. If you had a large robot (with a big battery), then you would have the room to do this. This is not an expensive sensor. I experimented some with the thought of getting delta-x and delta-y results and converting from rectangular to polar coordinates to calculate navigation angles, but found it difficult to interpret the results. If someone figures out a way to get reliable results, I'm sure our readers would love to know!

In conclusion, I have to say that this is a fascinating sensor with a variety of potential uses for robotics. It is also pretty inexpensive, which is also very nice. It draws quite a lot of current which limits its use in odometry to larger robots with larger batteries, or as a motion detector that can be turned on and off if we use a

Figure 5. Optical motion sensor system.



different lens that focuses further away than the 1-2 mm typically used for mouse movement detection. It is certainly worth a try since it is cheap and easy to use with our microcontrollers!

Figure 5 shows my final system. The Pololu Orangutan is perfect for this kind of quick test of an idea – it has an LCD display, a motor driver, piezo speaker, pins and buttons, and even a power switch. Doesn't that red glow from the ADNS-5050 optical sensor LED look cool?

The entire source code for this

project can be found on the *SERVO Magazine* website www.servomagazine.com under the article link. I used avr-gcc with the Eclipse IDE setup I've described in the past. Any avr-gcc (like Winavr) will work. Check out www.pololu.com for an Orangutan board and, of course, www.digikey.com for all of the components used in this project.

Keep building those robots and keep sending me your questions to roboto@servomagazine.com. I'll do my best to answer them! **SV**

LISTING 3. Main sensor board code.

```
int main(void)
{
    char xVal = 0;
    char yVal = 0;
    int16_t tempX, tempY;
    int16_t acumX = 0;
    int16_t acumY = 0;

    _delay_ms(100);
    InitLCD();
    InitMouse();

    /*
     * Get the Product ID (0x12) to show we're working.
     * Get the resolution, 0 = 500 cpi, 1 = 1000 cpi
     */
    xVal = ADNS_wr(PROD_ID,0);
    tempY = ADNS_wr(M_CTRLR,0);
    PrintVals((int16_t)xVal, (int16_t)tempY);
    _delay_ms(500);

    /*
     * Sample the sensor every 10ms and accumulate the vector
     * sum. We only have 8 bits, so if we move too fast we're
     * likely to lose some information.
     */
    while (1) {
        _delay_ms(10);
        xVal = ADNS_wr(MOTION,0x00); // To trigger DELTA reading
        xVal = ADNS_wr(DELTA_X,0x00); // 8 bit signed value
                                         // in X axis
        yVal = ADNS_wr(DELTA_Y,0x00); // 8 bit signed value
                                         // in Y axis
        if ((xVal != 0) || (yVal != 0)) { // If either are != 0
                                         // print
            tempX = (int16_t)xVal;
            if (xVal & 0x80) { // negative value
                xVal = ~xVal + 1;
                tempX = (0 - (int16_t)xVal);
            }
            tempY = (int16_t)yVal;
            if (yVal & 0x80) { // negative number
                yVal = ~yVal + 1;
                tempY = (0 - (int16_t)yVal);
            }
            acumX += tempX;
            acumY += tempY;
            PrintVals((int16_t)acumX, (int16_t)acumY);
        }
    }

    return 0;
}
```

An
easier,
more **reliable**
way to
'cut
the wire!'



Ready for wireless but unsure about the best path? Anaren Integrated Radio (AIR) modules offer:

- > Industry's easiest, most cost-effective RF implementation
- > Low-power RF solution
- > Virtually no RF engineering experience necessary
- > Tiny, common footprints
- > Pre-certified/compliant: FCC, IC, ETSI (as applicable)
- > Choice of modules based on TI CC11xx and CC25xx, low-power RF chips: 433MHz, 868MHZ (Europe), 900MHz, 2.4GHz

To learn more, write AIR@anaren.com, visit www.anaren.com/air, or scan the QR code with your smart phone.

ONLY \$999
FOR 10K OR MORE!



TEXAS INSTRUMENTS
MCU Developer Network



Anaren®
What'll we think of next?®

800-411-6596
www.anaren.com
In Europe, call +44-2392-232392

Available from:



EVENTS

Calendar

ROBOTS.NET

Send updates, new listings, corrections, complaints, and suggestions to: steve@ncc.com or FAX 972-404-0269

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: <http://robots.net/rcfaq.html>

— R. Steven Rainwater

JULY

4-10 RoboCup Robot Soccer World Cup

Istanbul, Turkey

All the usual robot soccer events including software simulations, small robot soccer, legged robot soccer, even humanoid robot vs.

human soccer. www.robocup.org

8-12 BOTBALL National Tournament

Orange County, CA

Student-built autonomous robots move black and white balls on a game board.

www.botball.org

11 RobotRacing

University of Waterloo, Ontario, Canada

Autonomous robot car races.

www.robotracing.org

12-17 AUVS International Underwater Robotics Competition

SSC Pacific TRANSDEC, San Diego, CA

Autonomous underwater robots must complete a

Build Your Robot The Way YOU Want!

With the Cross-link Robot Control System!

With this system, there's no need to write software! Plus, you get portability, connectivity, and reliability for one low cost! You can use ANY processor, ANY platform with ANY toolsuite, or run it directly from a PC or laptop!



The kit includes:

- 2CAN Ethernet Gateway
- CANipede Robot Control Module (RCM)
- TRENDnet N Pocket Router
- Cross-link/CAN Cables

Control Your Bot with an Android Powered Device!



Cross the Road
Electronics, LLC

www.crosstheroadelectronics.com



AP CIRCUITS

PCB Fabrication Since 1984

As low as...

\$9.95
each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com



course with various requirements. The contest is being renamed to the "International RoboSub Competition" this year; www.auvsifoundation.org/AUVSI/FOUNDATION/Competitions

25-29 K*bot World Championships

Las Vegas, NV

Student teams from countries around the world compete with autonomous and remote controlled robots built from K'NEX parts in multiple divisions. www.kbotworld.com

AUGUST

6 Chibots SRS Robo-Magellan

Moraine Valley Community College, Palos Hills, IL

A way-finding competition for autonomous robots that requires robots to navigate real world obstacles while locating orange cones.

www.chibots.org

7-11 AAI Mobile Robot Competition

San Francisco, CA

This year's events for autonomous robots include the Learning from Demonstration challenge, the Miniature Humanoid Obstacle Avoidance Challenge, and the Small Scale Manipulation Challenge.

www.aaai.org/Conferences/conferences.php

8-12 AUVS International Aerial Robotics Competition

Betty Engelstad Sioux Center, UND
Grand Forks, ND

This year's challenge requires advanced autonomous indoor flying. The organizers promise the contest "demands more advanced behaviors than are currently possible in any existing aerial robots. <http://iarc.angel-strike.com>

16-20 World Robotic Sailing Championship

Lübeck, Germany

Three classes of robot boats will compete: MicroTransat, Sailboat, and Design MicroMagic. This year's challenge is focused on testing control algorithms and boat performance.

www.roboticsailing.org

31 Homebrew Robotics Club Challenge, TABLEBot Phase II

Google, 1400 Crittenden Lane
Mountain View, CA

HBRC is holding a three-phase TABLEBot challenge in 2011. Phase I was on June 29, Phase II is August 31, and Phase III is October 26. Each phase test is a more complex behavior of table-top autonomous robots.

www.hbrobotics.org

STEER WINNING ROBOTS WITHOUT SERVOS!



Perform proportional speed, direction, and steering with only two Radio/Control channels for vehicles using two separate brush-type electric motors mounted right and left with our **mixing RDFR dual speed control**. Used in many successful competitive robots. Single joystick operation: up goes straight ahead, down is reverse. Pure right or left twirls vehicle as motors turn opposite directions. In between stick positions completely proportional. Plugs in like a servo to your Futaba, JR, Hitec, or similar radio. Compatible with gyro steering stabilization. Various volt and amp sizes available. The RDFR47E 55V 75A per motor unit pictured above. www.vantec.com

VANTEC

Order at
(888) 929-5055

AndyMark

Inspiring Mobility

Specializing in Unique Wheels,
Gearboxes, Aluminum Sprockets
and Drive Bases



6" Aluminum Dualie
Omni Wheel

Tri- Lambda Drive Base
Omni-directional drive
system kit.



Aluminum Sprockets

8" Mecanum Wheel



AndyMark, Inc.
sales@andymark.com

Toll Free: 877-868-4770

www.andymark.com

NEW PRODUCTS

BOOKS

***Robot Builder's Bonanza* Fourth Edition**

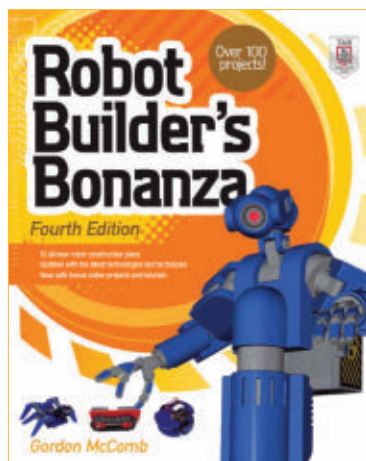
Fully updated with the latest technologies and techniques, *Robot Builder's Bonanza*, Fourth Edition includes step-by-step plans that take you from building motorized platforms to giving the machine a brain — and teaching it to walk, talk, and obey commands. This best-selling book is written by regular *SERVO Magazine* contributor Gordon McComb.

The aim of this richly-illustrated 700 page guide is having fun while learning how to construct sophisticated, fully autonomous robots that can be programmed from your computer. "RBB4" is packed with over 100 affordable projects, including 10 completely new robot designs. The projects are modular and can be combined to create a variety of highly intelligent and workable robots of all shapes and sizes.

Robot Builder's Bonanza, Fourth Edition includes coverage of:

- Parts, materials, and tools.
- Building motorized wooden, plastic, and metal platforms.
- Rapid prototyping methods.
- Drafting bots with computer-aided design.
- Building bots from found parts.
- Power, motors, and locomotion.
- Robots with wheels, tracks, and legs.
- Constructing robotic arms and grippers.
- Microcontrollers: Arduino, PICAXE, and the BASIC Stamp.
- Robot electronics and circuit making.
- Computers and electronic control.
- Remote control systems.
- Sensors, navigation, and visual feedback.
- Robot vision via proximity, light, and distance.
- Free online content, including videos and tutorials.

Robot Builder's Bonanza, Fourth Edition is published



by McGraw-Hill and is available through the *SERVO* Webstore at store.servomagazine.com.

For further information, please contact:

**Robotics
Universe**

Website: www.robotoid.com

PCB SERVICES

Shorter Lead Times for CAD Tool PCB123

Sunstone Circuits® — a printed circuit board (PCB) prototype solutions provider — is now offering shorter lead times for their circuit board design software, PCB123®. The shortened lead times now correspond with Sunstone's quickturn PCB lead times, and join a growing list of CAD tool product features which make the prototyping process even easier and more accurate from project concept to PCB delivery:

- Datasheet availability.
- Automated Bill of Materials (BOM).
- Expanded parts libraries.
- Seamless synchronization of layout and schematic views.
- Optimized graphics.

"Since September 2010, we have released a half million parts, upgraded the CAD tool's overall reliability, and refined and advanced our schematic editor capabilities," said Sunstone's CAM Team Leader, Michael J. Hebda. "With this new capability, we can build the PCB123 boards one day faster, so it's a win for our customers."

For further information, please contact:

**Sunstone
Circuits**

13626 S. Freeman Rd.
Mulino, OR 97042
Website: www.Sunstone.com

MISCELLANEOUS

Two-Channel Precision Joystick

ServoCity introduces a two-channel precision joystick which is able to operate two servos via the joystick.

This easy to use servo controller will run on a 4.8-6 VDC power source which simply plugs into the power port found on the front of the case. The lightweight yet durable 1/4" ABS plastic case is ergonomically designed to fit in the user's hand, but can easily be mounted to a flat surface using the mounting holes found at the top and bottom of the case. This unit is great for pan and tilt applications, robotics projects, and general servo operation.



Four Servo Recorder

The Servo Recorder also from ServoCity is a four-channel servo controller that is able to operate the servos individually or simultaneously. The internal memory of the servo recorder is able to store a recording up to three minutes and retains the motions of the servos, as well as the delay in between the servo movements. Once a recording is made, the user can select to play it back once by selecting "single" mode, or they can choose to repeat the recording by selecting "loop" mode. The unit doubles as a manual controller if the user wishes to operate the servos in real time without making a recording. The servo recorder is protected by a lightweight machined aluminum enclosure and is great for animation, robotics, and pan and tilt applications.

For further information on the previous two products, please contact:



10 mm output shaft for the strength needed for the most demanding large scale applications. With 347 oz-in torque, the HS-5765MH is tough enough to tackle projects from 1/5th on- and off-road vehicles to large high powered boats and giant scale aircraft. The estimated street price is \$100.



Digital Servo Field Programmer and Tester

Hitec also announces the arrival of the new digital servo field programmer and tester — the HFP-25.

Programming servos is now easier and more affordable. The field programmer has a lightweight, streamlined case and rechargeable NiMH battery pack, and is a must have for all hobbyists looking to program Hitec's HS-5XXX and HS-7XXX series of servos. The easy-to-read LCD screen, low battery warning, and both manual and automatic testing capabilities are just some of the helpful features which also include:

- Transmitter Pulse Width Measurement
- Receiver Voltage Measurement
- Direction of Rotation
- Speed of Rotation
- Fail Safe On-Off
- End Point
- Neutral Point
- Fail Safe Point Preset
- Rechargeable NiMH Battery Pack
- Overload Protection Setting*
- Servo Resolution Mode Setting*

* Applicable to Hitec's HS-7XXX series of digital servos only



The estimated street price on this product is \$74.99.

For further information on the previous two products, please contact:

Hitec

12115 Paine St.
Poway, CA 92064
Website: www.hitecrcd.com

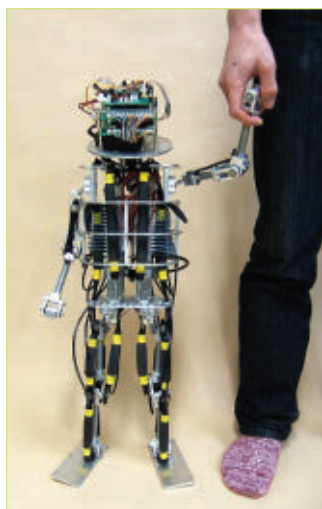
ServoCity

Website: www.servocity.com

HS-5765MH High Voltage, Digital, Metal Gear Giant Servo

Hitec introduces a class leading, high voltage, digital sport, metal gear giant servo called the HS-5765MH. When it comes to big RC projects, size does matter — that's why Hitec engineered the HS-5765MH. This high voltage cousin of their popular HS-5755MG is two-cell LiPo (7.4V) capable and features heavy duty metal gears with a

bots IN BRIEF



BABY STEPS FOR PROGRESS

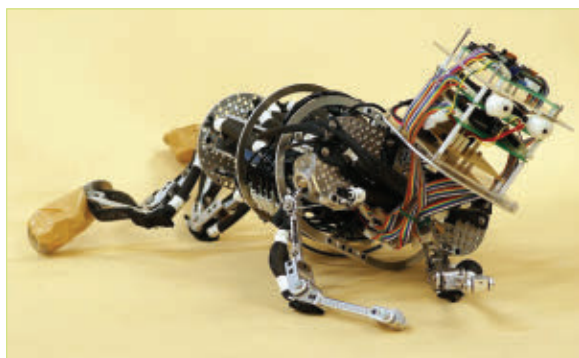
Osaka University's Hosoda Lab presented Pneuborn-7II and Pneuborn-13 — two musculoskeletal infant robots — at ICRA 2011. Their names play on the pneumatic muscles used as actuators throughout their bodies which can contract approximately 25% when compressed air is supplied. This type of actuator uses soft flexible materials, allowing the robots to interact with their environment — sometimes for several hours at a time — without risking mechanical damage or overheating.

Measuring the size of a seven month old infant, Pneuborn-7II was built to study the relationship between motor development and embodiment. It is 80 cm (31") tall, weighs 5.4 kg (11.9 lbs), and has 26 degrees of freedom actuated by 19 pneumatic muscles. Notably, the robot's spine has three pitch and yaw joints that allow it to rotate, flex, and extend. It is fully autonomous and contains a microcontroller, battery, air valves, and an air source (compressed CO₂ cartridge bottle). During long experiments, air can be provided through an external compressor.

The researchers implemented a learning algorithm based on central pattern generators with an optimization method which was able to generate successful crawling forward motions.

They were able to accomplish this despite the robot's lack of sensors or sophisticated artificial intelligence. Central pattern generators (or CPGs for short) are a type of neural network that are often used in robotics to create rhythmic motions that are especially useful for simple locomotion such as crawling or rolling over.

Pneuborn-13 models a 13 month old infant, and was developed to study the effect the musculoskeletal structure itself has on the emergence of bipedal walking. As a result, its 18 pneumatic muscles are concentrated in its ankle, knee, and hip joints. It measures 75 cm (29.5") tall, weighs 3.9 kg (8.5 lbs), and has 21 degrees of freedom. Like Pneuborn-7II, it is wholly autonomous and has a similar skeletal structure, but it doesn't have an actuated spinal column. The robot is able to hold a standing posture and can make stepping motions.



A BIRDBRAIN IDEA

Carnegie Mellon University developed a robot to keep tech-savvy students constantly interested and came up with Finch — a white, plastic two-wheeled bot that resembles a bird. At a price of only \$99 (with discounts for ordering in quantity), young scientists can program it to speak, dance, and even draw pictures. Finch has a three-axis accelerometer, temperature, bump, and light sensors, LED lights, and speakers.

The eventual goal is to allow every student to adopt one and take it home for assignments. The company BirdBrain Technologies produces and sells Finch, and has developed lesson plans for teachers as well as giving them the option of uploading their own ideas. Classroom testing has been tried in high school, university, and after-school programs.

bots IN BRIEF



BOT ON A BUDGET

Tim Payne — the creator of the bipedal humanoid PROTO-2 — has developed an open source robot that can be built on a budget. It's called POLYRO (oPen sOurce friendly RObot), and makes extensive use of Willow Garage's TurtleBot as its mobile base. TurtleBot takes advantage of low cost components like the iRobot Create and Microsoft Kinect sensor, allowing it to autonomously map and navigate its environment. What POLYRO brings to the table is some much needed personality; it is

primarily designed for human-robot interaction and has a humanoid upper body. POLYRO stands 99 cm (3'3") tall and weighs 8.6 kg (19.5 lbs) without its netbook. Its two arms have three degrees of freedom apiece, its head nods, and its stereoscopic eyes can blink. It uses a total of 11 Robotis Dynamixel servos in its current design, two USB webcams for eyes, and a netbook PC running Linux, but together with all the miscellaneous parts it still adds up to less than \$2,000. The additional degrees of freedom will create more expressive gestures, and additions like speakers and microphones would allow it to use speech recognition and synthesis. It could, for example, serve as a guide robot around university campuses. In the meantime, you can find extensive documentation at Payne's Instructables post.



CROWD PEAS-ER

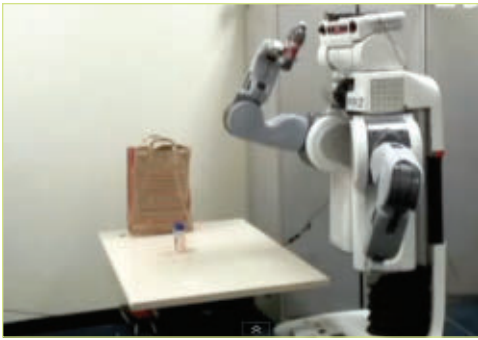
Anyone who has ever seen a Black Eyed Peas performance knows that will.i.am. is into robotics and he took on Dean Kamen — creator of the Segway — during the FIRST 2011 competition. The organization (which stands for "For Inspiration and Recognition of Science and Technology") was founded by Kamen in 1989. Following the opening ceremonies of championships, will.i.am and Kamen went head-to-head in a FIRST Tech Challenge match. The FTC class of competition features smaller bots than the FIRST Robotics Competition that is the headliner event. Kamen and will.i.am acted as honorary coaches in the FTC exhibition, with Kamen backing team 2859 from McLean, VA, and will.i.am behind team 3509 from Folsom, CA (nicknamed "The Dirty Bots" in his honor).

When the California team gave the singer a turn at the controls, he performed admirably, says sophomore Millun Atluir, who is competing in his second year of FTC. "He did pretty well," Atluir says. "His driving blocked the opponents so we could get the magnetic baton." Indeed, when the final buzzer sounded, will.i.am's team had defeated the competition 73-9.

DMBH AND DMBHR

A team of researchers at the Institute of System Information and Control at the National Kaohsiung First University of Science and Technology, Taiwan, have developed the university's first full-scale humanoid. Nei Dau (a.k.a., Dynamic Motion Balance Humanoid Robot, or DMBHR for short) stands 130 cm (4'3") tall but weighs just 9.6 kg (21 lbs). Despite its size, it makes use of Robotis RX64 and AX12 servos (24 in all) for its 20 degrees of freedom (two legs x6, two arms x3, head x2). It has two cameras and makes use of standard sensors to keep its balance (gyro, accelerometer, and force sensors in the feet). The robot previously competed at RoboCup 2010, and competed for Team Kaobotics in the Adult Size League at RoboCup 2011.





CHECK IT OUT

Normally, when a robot wants to pick something up that it's never seen before, it either has to download a 3D model of the object, make its own 3D model and analyze it, or be trained by a human on the right way to grip. Unfortunately, none of these things are really practical to do in the fast paced world of grocery checkout lines.

Researchers at Stanford University have figured out that in order to pick something up, all you really need to know is whether a piece of it has the same basic shape as the shape of your gripper. If it does, then you can most likely grip it fairly well, and experimentally the success rate is better than 90 percent. Best

of all, you can extract this shape information from one simple (and quick) 3D scan, even if you've got a big cluttered pile of stuff. Once the robot has picked up an object, it holds it up to its cameras to scan for the barcode, adds it to your tab, and bags it for you. While you probably aren't going to see PR2s down at your local Trader Joe's, the code that's being developed here could conceivably find its way into some kind of grocery robot in the future, or even into a robot that picks up and puts away stuff in your house.

NASA GETS A SCOOP ON ASTEROID

Humberto Campins — the University of Central Florida professor who discovered ice on asteroids last year — is now working on a new NASA project. The \$8 million OSIRIS-REx is unmanned and will be launched in 2016 to map and take pieces of asteroid 1999 RQ36 with its robotic arm. After returning to Earth to turn over the samples, it will then go into orbit around the sun for other experiments. NASA is hoping to gain knowledge about our planet and track asteroids in the future.

The mission is a first of its kind. The actual flight to the nearby asteroid will pose challenges because asteroids have unusual gravity fields and can rotate much quicker than planets. Navigating their space vehicle to land on this type of asteroid — millions of miles away from Earth — and scoop up a sample of “primitive” space rock also will be a first for the team. While Campins is leaving the navigation to others on the team, he will work with lead investigator Michael Drake (from the Lunar and Planetary Laboratory at the University of Arizona) on choosing the best spot on the asteroid for obtaining the sample and what this sample will tell us about the origins of life on Earth. The non-manned mission could launch as early as 2016.

Money for the project will cover the cost to circle and analyze the asteroid for a year with an array of special instruments to help select the optimal sampling spot. That's where a mechanism will scoop up the sample and place it in a capsule. That capsule will later come back to Earth, splashing in an ocean with the help of a parachute. Funds also cover the design of the instrument that will lasso the sample and bring it back to Earth.

“The even harder work will begin once we have our sample,” Campins said. “We will spend at least two years going over every piece of information we gather.”

The target — asteroid RQ36 — is about 600 yards in diameter and comes relatively close to Earth. The asteroid has even been designated as “potentially hazardous” because there is a one in 1,800 chance that it could slam into Earth in 2170.



‘RUTH’LESS BEHAVIORS

Ruth — a Robotized Unit for Tactility and Haptics — has been used by Ford in their research center in Germany to help develop better vehicles. Engineers taught her to “feel” what humans like with sensitive fingers and so that she can predict what new components will be desirable to them. Ruth has worked on the operation of switches and textures of material. The center claims the recent attempt at developing a better steering wheel ended up with a 92% accuracy of judging what humans would prefer.

Ford engineers have recently been using Ruth to help design the optimal steering wheel by comparing the robot's measurements with detailed market research into customer perceptions of quality, such as the softness of leather and foam combinations.

ARMED AND READY

Fraunhofer has developed a sensor system that — with its remote — can move a robotic arm. Its technology is so advanced that it can catch a ball or move when the remote does. Algorithms help the technology detect movement and the European group hopes that future applications will include service robots and prostheses. The arm was unveiled at a Nuremberg trade fair in June.

Researchers presented an industrial robotic arm with six joints, at the end of which was a catcher. Visitors could control the arm using a handheld input device. When they move the hand holding the device, the robot emulated their movement. "The input device contains various movement sensors, also called inertial sensors," said Bernhard Kleiner of the Fraunhofer Institute for Manufacturing Engineering and Automation IPA in Stuttgart, who led the project. The individual micro-electromechanical systems themselves are not expensive. What the scientists have spent time developing is how these sensors interact. "We have developed special algorithms that fuse the data of individual sensors and identify a pattern of movement. That means we can detect movements in free space," summarized Kleiner.



ANDROID GOES GREEN

If you've been following the news coming out of Google I/O recently, you know that the Android OS is now being used to control lots of hardware besides phones and tablets. Android has been spotted running laptops, appliances, TVs, and more. And now, a particularly interesting application will allow that beloved little green robot to control a vegetable garden through the Cloud.

Brilliant Service (a Japanese company) has created a solution for a computer-controlled growing system known as a plant factory which they are calling Farmbox. These self-contained ecosystems are offered as a solution to the woes of modern agribusiness: the rising costs of quality produce, use of polluting fuels in both its production and transportation, dependence on climate, and the presence of dangerous chemicals. The solution brings vegetable cultivation into the home in the form of the plant factory which they see as becoming something of a household appliance. Environmental factors such as lighting, water, and air temperatures, nutrient, and oxygen levels will be regulated and maintained by an app on an Android phone. The app will upload data on your growing conditions to the Cloud where it can be accessed by other growers who will then be able to duplicate identical conditions in an entirely different location. Through this data-sharing system, the developers hope to achieve their vision of allowing anyone, anywhere, to grow good, healthy vegetables, effortlessly and inexpensively.



PLANT GUARD CUTE AS A BUG

Have you ever had trouble taking care of your houseplants? Well, the Korean Advanced Institute of Science and Technology (KAIST) has developed a simple "robot" just for you. They're calling it a "plant guard robot," and you stick it in your potted plants to monitor the soil moisture in real time. Apparently, its for plants that require regular care (not plants like cacti). If the water level falls below a certain level, a red LED will blink and the cricket will make a noise. If all's well, a soothing blue LED pulsates. Retail price will run about 9900 KRW (\$8 USD). In the future, they want to include other features like specializing it for certain plants or for detecting radiation levels or other harmful environmental factors. You can probably find something similar in stores, but they're probably not this cute!



Cool tidbits herein provided by Evan Ackerman at www.botjunkie.com, www.robotsnob.com, www.plasticpals.com, and other places.

COMBAT ZONE

Featured This Month:

Features

28 **MANUFACTURING:**
The Robotist's Guide to Plastic

by Matthew Spurr

30 **BUILD REPORT:**
Building a Better Beetle Beater Bar – Part 1

by Pete Smith

33 **CARTOON**

Events

32 **July 2011**
Upcoming Events

32 **EVENT REPORT:**
Robot Rumble

by Pete Smith

www.servomagazine.com/index.php?/magazine/article/july2011_CombatZone

MANUFACTURING

The Robotist's Guide to Plastic

● by Matthew Spurr

Plastic has become an integral part of our society. It holds our food, is found in our automobiles, and is even on and in our bodies. It also makes a great building material for robotics. Plastics as a building material category have many qualities which makes them appealing, such as cost, ease of forming/bending, ease of cutting/drilling/tapping, weight, and relative strength. Plastics are also very confusing; they come in a wide variety, use 13 syllable chemical compounds for the name or a seemingly random collection of letters for an abbreviation, and determining which plastic is best for your application can be difficult, until now.

In this article, we'll cover some of the best and most common plastics suitable for robot construction, what makes them good and bad, and where you can find examples of such plastics in everyday life. Each plastic has its advantages and disadvantages. I have personally

used nearly every plastic that will be listed here with varying results. In some instances, my decision has worked perfectly and in other cases, my decision was less than ideal. It should be noted that we are not attempting to discuss the seemingly limitless number of plastics commercially available.

The Basics

Plastic is a generic term for a synthetic or semi-synthetic material that can be formed when soft and then hardened. Plastics can be melted and pooled into two categories: the plastics that can be melted and pooled, and the plastics that can't. Basically, some plastics become soft and/or melt when sufficient heat is applied. These plastics are called thermoplastic. Other plastics harden with sufficient heat and are referred to as thermoset. The chemists who invented the various plastics were very smart, but the names they

came up with aren't all that imaginative.

Phenolics

Phenolic is a name for a plastic which contains phenol (C₆H₅OH) (again plastic inventors weren't the most original). Phenolics are thermoset plastics. The first fully synthetic plastic ever produced was a phenolic called Bakelite. Phenolics are typically brittle and relatively expensive; however they have great electric and heat insulating qualities which is why they are commonly used in circuit boards.

ABS

ABS is tough impact resistant plastic found quite commonly in the automotive industry. It is typically used for the interior trim pieces in your car. You may have had your first experience with ABS as a child playing with LEGO® bricks. It is more costly to produce than other materials and it can produce an intensely hot flame once ignited. It should also be noted that ABS has different grades that are based upon the impact qualities and the intended forming methods.

Polyethylene (PE/PET)

Polyethylene (and polyethylene terephthalate) is one of the most common plastics you will encounter. In fact, you could be wearing it or drinking from it right now. PET in fabric form is something you may have heard of — polyester. PET is also common in solid form and is used in soda bottles, detergent containers, milk jugs, plastic bags, etc. It is one of the cheapest plastics and offers good abrasion resistance and impact strength. It is a thermoform plastic, so it is easy to bend into most any shape your heart desires.

UHMW

Ultra-High Molecular Weight

Polyethylene is a subset of the previously mentioned polyethylene. I broke it out in a separate section because of its frequent use in robotic applications. UHMW has excellent wear resistance and a low coefficient of friction (COF) which lends itself to being used frequently as a bushing material. It also has excellent impact resistance which makes for a great frame material. Working with UHMW is easy, as you can use standard wood and/or metal working tools to cut, drill, and tap it. UHMW can be held together with wood screws or tapped for use with machine screws. UHMW is found in artificial joints, as a wear plate at the bottom of hockey rinks, and in cutting boards.

Acetal/Delrin

Delrin is a great material for areas where you are expecting to have parts moving past each other. Delrin has impact qualities close to ABS, but has a lower COF. Delrin is found in a wide variety of applications, such as gears and bushings, and is also used for several parts in paintball and airsoft guns. Acetal has fair impact qualities, but from experience is probably best left for friction components and not used for frame material.

Acrylic/Plexiglass

Acrylic is a great plastic if you are more interested in appearance and cost than taking an impact. Acrylics are commonly used as a cheap and more shatter resistant replacement to glass. Acrylic will still shatter under load — especially an impact load — and is not very scratch resistant. It is much cheaper than the more impact resistant polycarbonate. Due to its strength and its ability to be produced in large sheets, acrylic is used for the large viewing windows in aquariums. You probably also drove by acrylic the last time you were in your car, as it is used frequently to

make large store signs.

Polycarbonate

Polycarbonate is a plastic that offers exceptional impact resistance and light transmission, i.e., it's clear. Polycarbonate is found in laminated form to produce the bullet-proof (bullet-resistant) windows at the bank. Polycarbonate is the plastic of choice, and the only one I recommend for use in combat arena windows. Polycarbonate is very expensive and is not very scratch resistant. For combat arena usage, it is recommended to apply a thinner, sacrificial layer to the interior of the arena that can be replaced once it is scratched to the point where viewing becomes an issue. Polycarbonate is not a brittle plastic, so it is possible to form without adding heat.

PVC

Polyvinyl Chloride (PVC) is a good plastic commonly found in the construction industry. It is cheap to produce and is easy to bond. PVC is a controversial plastic because as certain additives off-gas they may produce toxins, especially if burned. It is most notably found as PVC pipe for plumbing and irrigation. It makes a good robotic material because it is readily available at any major hardware store and is easy to form into a wide variety of shapes. PVC is also commonly found in window and door frames, wire insulation, shower curtains, and in components used for IVs.

Wrap-Up

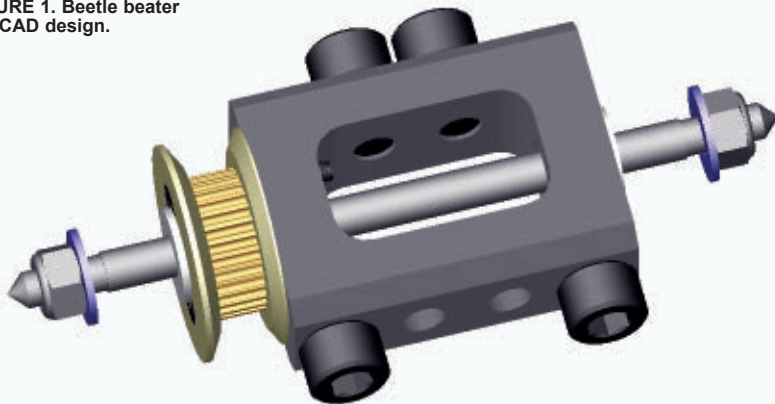
Plastics come in a wide variety of shapes, colors, opacity, and material qualities. Hopefully, you've found this article useful in selecting which material you'll be using next in your robotics applications. When it comes to plastics, there is no replacement for experience, so get out there and try some. Feel free to get back with me with your results. **SV**

BUILD REPORT: Building a Better Beetle Beater Bar!

Part 1 – The Beater

● by Pete Smith

FIGURE 1. Beetle beater bar CAD design.



Beaters are a popular adaptation of the conventional drumbot design. The circular drum is replaced with a thresher or beater bar. My beetleweight drumbot Weta, God of Ugly Things, was originally fitted with a 6061 aluminum drum that had M8 socket head screws as

teeth. This worked well enough, but the soft 6061 (it was cheap and easy to machine) was not up to the task of retaining the teeth and by the end of the event, it was definitely worse for the wear. The event also showed I needed a bigger hit and a better bite.

A higher rotational speed gives a bigger hit but it also reduces the bite. One way around this is to use a beater bar design. This article will show you how a simple beater bar can be created.

First thing is to design the bar assembly and calculate its weight. It's too easy to design an effective weapon and then

find out it's too heavy to be used on your bot!

Personally, I design using SolidWorks, and it includes a weight calculator. Other CAD tools can be used, or even pencil, paper, and a little math will get you very close to the finished weight.

The chosen design can be seen in **Figure 1**. An aluminum block uses needle roller bearings to revolve on a 5/16" diameter ground titanium shaft. The teeth are M10 socket head screws. A 30 tooth 3 mm HTD pulley with UHMW flanges (down from 40 tooth in the original to increase RPM by 33%) is mounted on one end. It is designed to work in a bot with the axle centerline 1.5" off the ground.

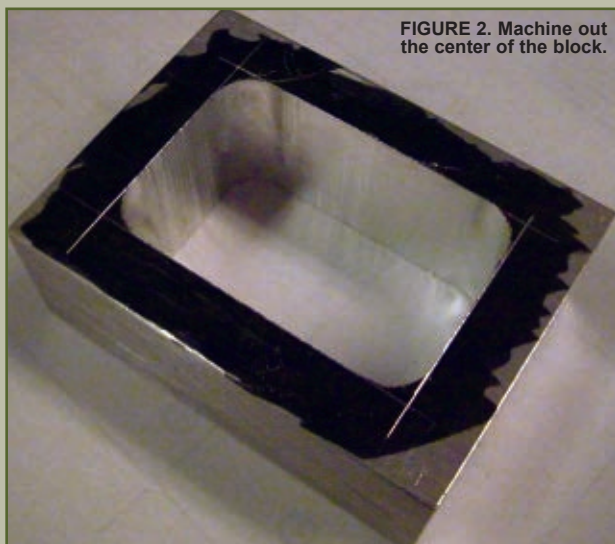
All drawings are available as a download from *SERVO* through the article link or through www.kitbots.com under the 3 lb beetle kit section.

The block is cut from a 2" x 1" bar of 7075 aluminum. This conveniently gives two of the three required outside dimensions without further machining. The block was cut roughly to length using an abrasive cut-off saw, and the ends were machined square and to length in my mill.

Safety Note: Milling machines are dangerous if not used correctly. Remove jewelry and any loose items of clothing, and ALWAYS wear safety glasses.

The center cut-out was marked carefully and then machined out

FIGURE 2. Machine out the center of the block.



using a 1/2" diameter center cutting end mill. I rough-machined to within 1/16" of the finished sized and then carefully did a finish cut to the scribed lines (**Figure 2**).

The next job is to drill, then ream the axle bearing mounting holes. In an attempt to keep the two holes as accurate and as in line as possible, I machined both in a single operation.

First, I carefully positioned the mill head over the scribed center point, then milled a 3/8" hole through both. I followed this with a 15/32 drill, then finished with a 1/2" reamer (**Figure 3**). If you mess up (as I have on occasion) and end up with the holes a little off-center, don't panic as you can further machine the block so the holes are on-center. What you need to remember is the block is no longer 1" thick or 2" wide, and compensate appropriately. Do not be tempted to think the off-center position won't matter. It will make balancing the bar much harder if you do not deal with it now.

There are only two teeth on either side of the beater. Staggering them (as shown in the **photo**) gives a potential bigger bite as the bar can turn almost a full revolution before a tooth first hits the target. However, we still need to have four holes in each side since the outer holes are blind and will require deeper holes to allow them to be easily tapped.

If I did not drill the same holes to the same depths on both sides, it would make balancing the bar much harder. The four holes were carefully scribed, center-punched, and then given a start with a smaller sized drill bit. I find this helps reduce the bigger finish size drill bit from wandering (**Figure 4**). It's also important to control the depth of the outer holes to prevent going too far and possibly affecting the axle holes or the screws that will be used to mount the pulley.

The holes were given a little chamfer with a countersink bit and then were tapped M10 using a



FIGURE 3. Drill, then ream bearing locations.



FIGURE 4. Drill teeth locations.

regular, and then plug or bottoming tap to get the thread complete almost to the bottom of the blind holes (**Figure 5**). I only tapped two holes on either side, but in hindsight it would have been better to tap all four as this would make it easier to swap teeth around if the beater gets damaged in a fight. One could also put a button head screw in the empty holes — if weight allows — increasing the mass of the beater.

The chamfers on the edges beside the teeth are added last using a chamfer bit in the mill. These reduce the chance of the soft aluminum hitting the target rather than the hard steel of the teeth.

The teeth are then screwed in (I use McMaster part 91290A510) and the needle roller bearings (I use part# BA-57Z from www.bearingsdirect.com) are pressed in using a bench vise (protect the bearing by padding the vise jaws or use a set of soft jaws) so that they are flush with the face of the beater.

The beater can now be balanced by running a section of the titanium rod that will be used as the axle (McMaster part 89055K341) and spinning the beater while holding the rod horizontal (similar to how it will

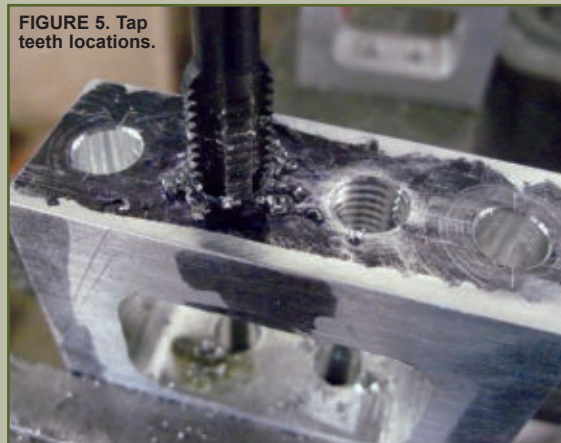


FIGURE 5. Tap teeth locations.

spin in the finished bot). If the beater always stops or returns to the same position, it means that the side at the bottom is slightly too heavy.

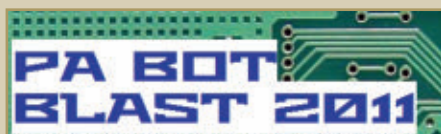
This can be remedied by removing a little metal from that side of the bar. The bearings are remarkably friction free and even a very small difference will be obvious. Don't remove too much metal at one time, and always protect the bearings from metal filings, etc., by taping over them with painter's blue masking tape. Repeat this process until the beater no longer stops spinning in the same place. It is now balanced and will run much smoother in the finished bot. This helps the bot drive since the wheels are not bouncing around so much, and also reduces vibration damage to the other components.

In a future article, I will show how to create the axle, drive pulley, and flanges. **SV**

EVENTS

Upcoming Events for July 2011

Bot Blast 2011 will be presented by Team Dreadfully Wicked Robots on July 16th at the Columbia Mall in Bloomsburg, PA. Go to www.botblast.us for more information.



presented by Carolina Combat Robots in Gastonia, NC on July 23rd. Go to www.carolina

Schiele Museum Clash Of The Bots 2 will be

combat.com for more information. **SV**



EVENT REPORT: Robot Rumble

● by Pete Smith

The Museum of Life and Science in Durham, NC held its fourth annual "Robot Rumble" on Saturday April 16, 2011.

Carolina Combat (www.carolinacombat.com) and Kitbots (www.kitbots.com) combined to put on a show as part of the event. Our involvement has grown over the last three years we have attended, and now we had a whole hall to ourselves. This allowed us to set up the bot hockey arena so that the crowd could get to all four sides, an insectweight arena, and a display

area of combat bots and Kitbots products.

The bot hockey matches were held with two members of the public and one experienced driver on each "team" (**Figure 1**). The audience members were shown how to operate the bot and then they would take part in a two minute match. This format worked pretty well as the one skilled driver on each team could make a match of it for the other spectators while the other drivers usually just added to the general mayhem.

Matches were closer than in past events as Kitbots fielded a brand new set of hockey bots specially built for the event (and finished just the night before!) This — combined with Andrew Smiths' skill — meant most rounds went to the Kitbots team this year. However, we did



not have it all our way as Chuck Butler skillfully made a breakaway for a goal for Carolina Combat (**Figure 2**).

We ran two minute matches for half an hour every hour, and used the other half hour to recharge the batteries (**Figure 3**; the pizza was to recharge the drivers!), run demos using Beetleweight spinner Weta, God of Ugly Things to beat up plastic bottles and soda cans in the small arena, and to talk with passersby

FIGURE 1.



FIGURE 2.



about the sport and the bots that were on display.

Good crowds attended throughout the day and as a result, at least one new convert to the combat side of the sport was made.

The bots proved generally reliable with only a shortage of chargers being a problem for Kitbots, and then a burnt out motor on one of Carolina Combats bots late in the day (they had a spare bot), causing

a few delays.

Thanks to the Museum staff and our families for helping make it all possible. I think it's a sure bet we will be back next year with an even bigger and better show. **SV**

FIGURE 3.



Melty Brains

by Kevin Berry

**Elementary School
Car Loop**

**My Child Is An Honor Student
At Snooty Elementary**

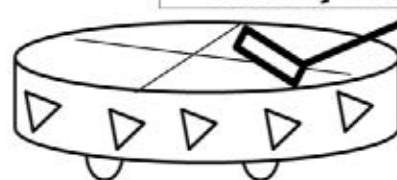
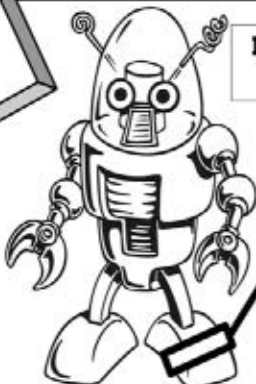
**My Delinquent Can Beat
Up Your Honor Student!**



**Robot Club
Car Loop**

**My MiniSapien has Levered
Posable Arms**

**My Beetleweight Can
Beat Up Your Sapien!**



2011 VEX Robotics World Championship

by R. Steven Rainwater

This year, the VEX Robotics World Championship moved from Dallas, TX to Walt Disney World's ESPN Wide World of Sports complex near Orlando, FL. The event was bigger than ever and, in addition to all the matches between more than 500 VEX teams from around the world, VEX played host to several other events as well. If you were one of the 10,000+ participants this year, you also got a chance to watch the BEST championship, the Coast Guard Academy's AROW water robotics competition, and the official roll-out of the new robotics merit badge by the Boy Scouts of America.

Given the location, it shouldn't be surprising that Mickey, Donald Duck, and assorted other Disney characters made random appearances during the event. Adding to the craziness this year were stilt walkers, a graffiti artist, a DJ, confetti canons, and occasional pyrotechnics.

Artist Kari Byron — best known as one of the MythBusters and host of Head Rush — was Master of Ceremonies. She spent lots of time on stage announcing matches, introducing speakers, and handing out awards. When she wasn't on stage, she was talking to robot builders, signing autographs, and posing for photos with teams. I think she may have been the most popular host in the several years I've been going to this championship event.

The winning high school alliance included the Green Egg Robotics Club from Massachusetts, W.A.S.A.B.I. 2 from Washington, and Simbotics from Ontario. The winning middle school alliance included Sichuan Chengdu Longjiang Road Primary School and Shanghai Luwan Teenagers Activity Center, both from China. Massey University from New Zealand took the College Championship title. The coveted Excellence Award went to the VEXMEN which was made up of the NightCrawler team from Downingtown Area Robotics in Downingtown, PA; the Cheesy Poofs from Bellarmine College Prep in San Jose, CA; and Massey



Kari Byron of MythBuster's fame.

University in New Zealand.

The VEX judges stayed busy evaluating the hundreds of teams, and there were plenty of other awards to go around for robots, teams, mentors, teachers, volunteers, and partners.

VEX participants showed their enthusiasm by cheering for their favorite teams and by competing for the most outlandish hair style. Blue hair and green hair were equally popular this year, but mohawks were a less common sight than in 2010. VEX teams couldn't match BEST for the most spirited supporters, however. Despite being a smaller crowd, BEST supporters consistently matched or exceeded the noise level of VEX supporters. Next year, they need to bring a sound level meter and award a prize for the team with the loudest supporters!

Meanwhile, on the other side of the sports complex, the United States Coast Guard Academy was busy holding matches between teams in the AROW water robotics competition. AROW robots were a combination of a styrofoam flotation platform and VEX hardware that drove one or more propellers. The result was a teleoperated boat that completed a series of tasks simulating common Coast Guard activities such as rescuing vessels, intercepting drug smugglers, or responding to oil spills.

The Boy Scouts of America continued their tradition of equipping boys for a useful role in society by promoting

interest in technology. Around 25 scouts completed the requirements for the newly announced robotics merit badge at the event. To earn the badge, a scout must design, build, and demonstrate a working robot. Each scout must also demonstrate an understanding of sensors, actuators, mobility, programming, and human-robot interaction. Some of the robots demonstrated at the event included a trash collection robot and a robot carrying a marker that could write out words by rolling across a large sheet of paper. The writing robot spelled out KARI for Kari

Byron when she stopped by the scouting area to meet the boy scouts.

As always, one of the highlights of the VEX Robotics World Championship is the sneak peak at next year's event. The new game will be called Gateway. Like past games, the new challenge involves robots moving blue and red objects around on the playing field, but the difficulty level has been increased. Next year's championship — which will be held at the Disneyland resort in Anaheim, CA — should be more challenging than ever. **SV**



Last minute preps.



VEX T-shirt canon robot fires into the crowd.



Awaiting the judge's decision.



VEX robot with an image of Charlie Sheen on it.

Photos continued on next page



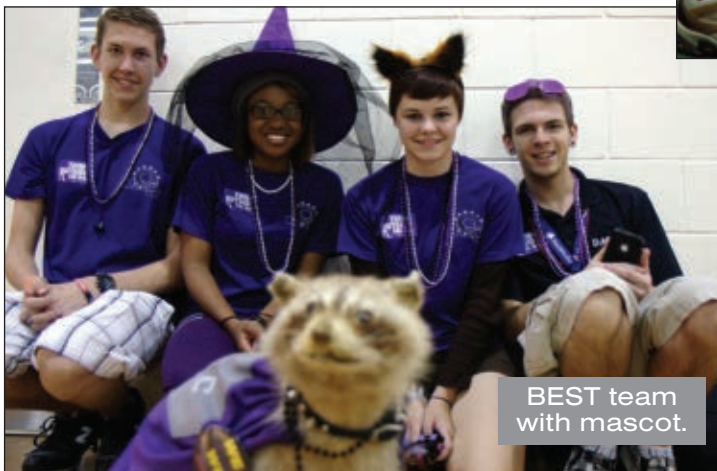
VEX winners.



Robot at one of the BEST team booths.



Ref checks BEST bot.



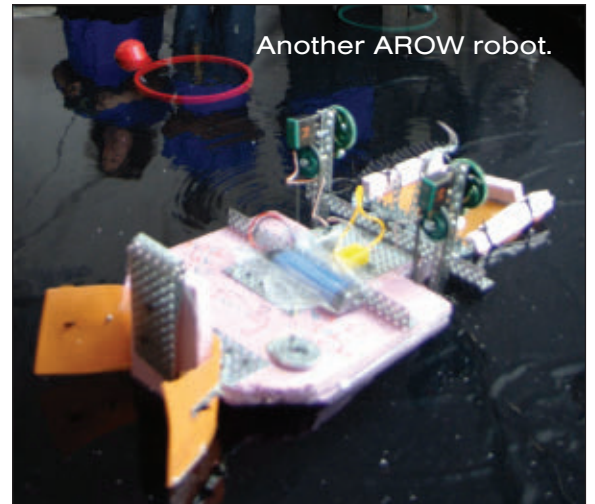
BEST team with mascot.



BEST team supporters



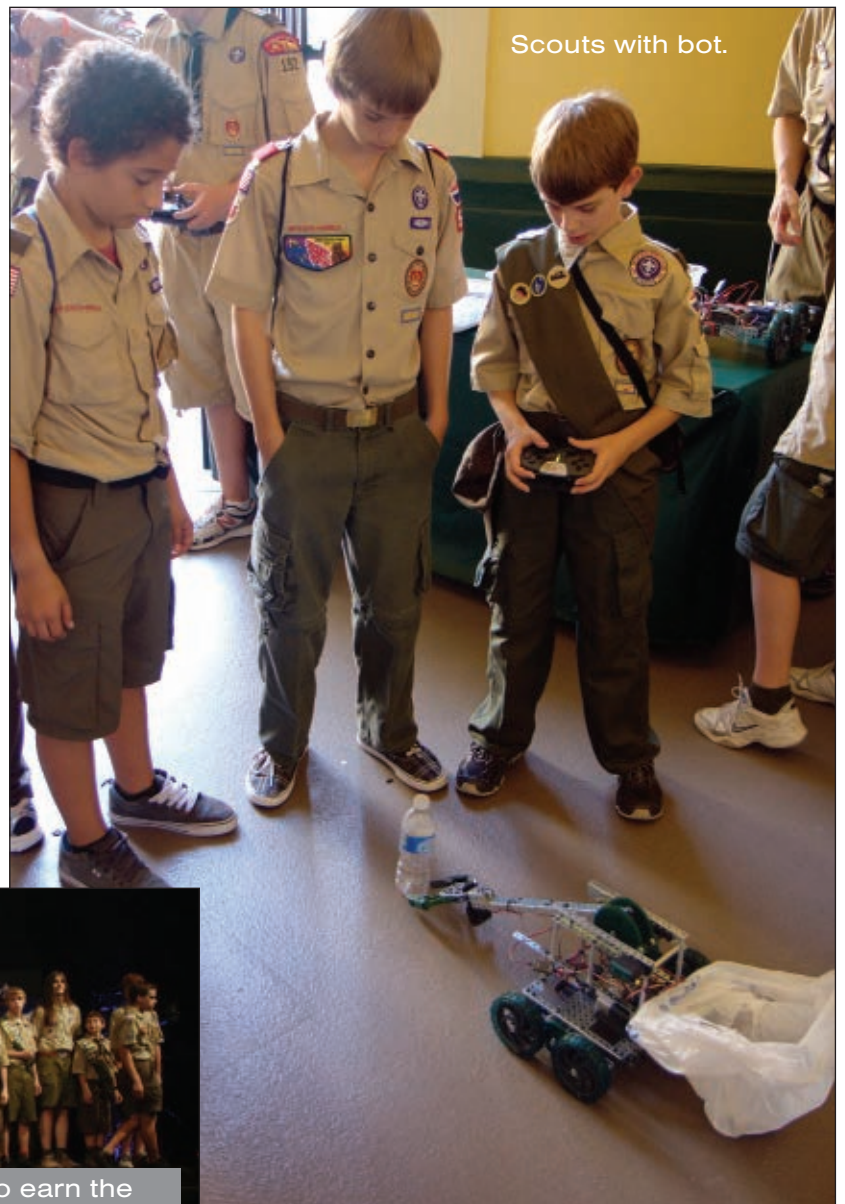
Watching
CG AROW.



Another AROW robot.



Boy scouts explain merit
badge requirements to Kari.



Scouts with bot.



First ever scouts to earn the
robotics merit badge.

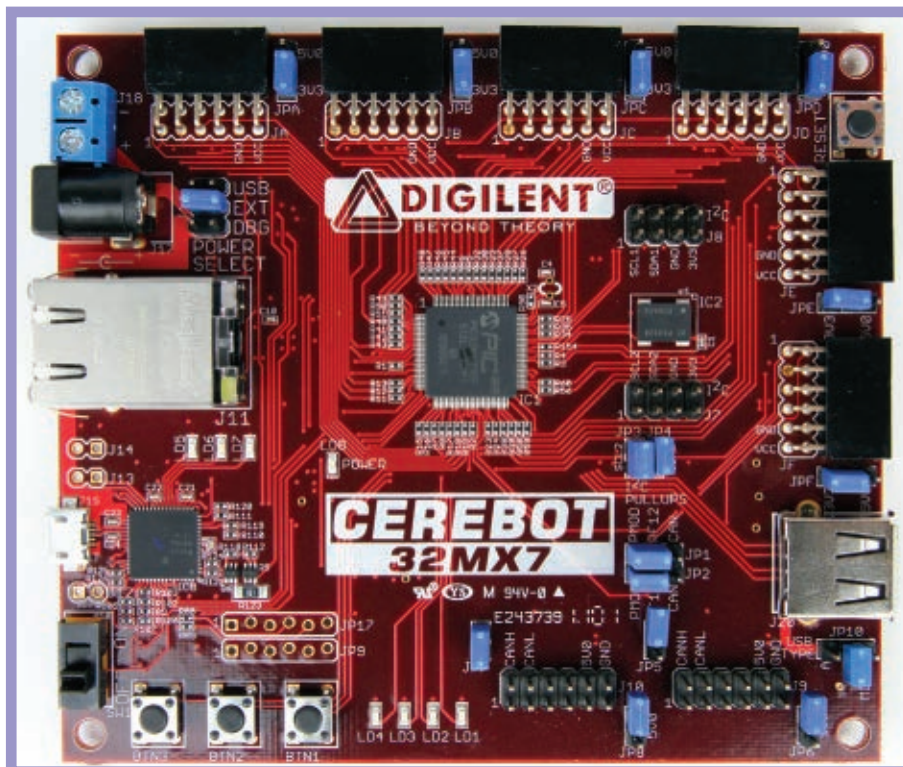
Meet the Swiss Army Knife of Robotics: the Cerebot 32MX7

by Fred Eady

As a young Boy Scout, my prized possession was my Swiss Army knife. There was no can or bottle that I could not open, and nothing existed that could not be screwed down, clipped, cut, bored, or sawed. These days, the tents, specialized knives, and mosquitoes have been replaced by soldering irons, microcontrollers, and C compilers.

The Swiss Army knife we are about to discuss has no blades, saws, screwdrivers, or can openers. The numerous blades that make up a typical Swiss

camping knife have been replaced by microcontrollers, specialized ICs, EEPROMs, transistors, LEDs, voltage regulators, resistors, capacitors, crystals, and firmware in nonvolatile memory. With a little help from a suitable mechatronic device, our silicon Swiss Army knife “blades” can also open a can, drill a hole, or drive a screw.



An FR-4 Glass/Epoxy Swiss Army Knife with Silicon Blades

The Digilent Cerebot 32MX7 illuminated by Paul C. Buff flash units in **Photo 1** is based on the 32-bit Microchip PIC32MX795F512L microcontroller. The abundance of right-angle female connectors, jumpers, and male headers suggests that every peripheral subsystem contained within

PHOTO 1. Whether it be robotics, remote control, telemetry, Internet services, CAN networking, or an I/O intensive application, the Cerebot 32MX7 is designed to plug physically and logically into the center of the action.

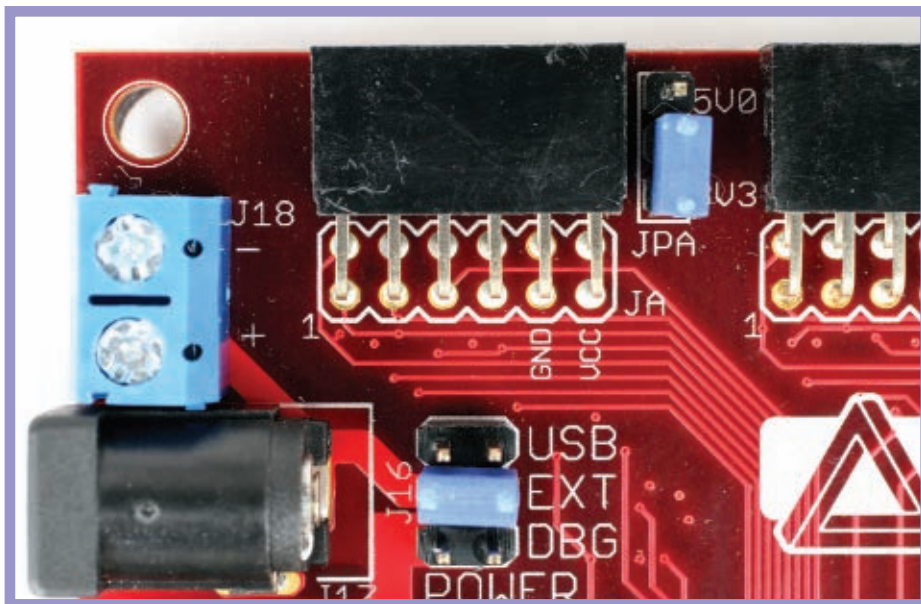
PHOTO 2. This is an overhead view of the Cerebot 32MX7's JPA Pmod portal which is configured to power a 3.3 volt Pmod or user-designed device. Note the POWER SELECT header (J16) is jumpered to power the Cerebot 32MX7 using external power. The external power supply can be connected at J17 or J18.

the PIC32MX795F512L's silicon is exposed in the Cerebot 32MX7 design. The 32MX7's silicon "blades" are made up of Pmod I/O portals, a USB host/device/OTG subsystem, a 10 Mbps/100 Mbps Ethernet subsystem, two independent CAN networks, and a pair of I²C interfaces.

Cerebot 32MX7 Pmod Portals

The Cerebot I/O complex includes six powered I/O portals designed to accommodate the growing family of Diligent Pmod (Peripheral Module) accessory modules. Each Pmod portal consists of eight lines of PIC32MX795F512L I/O and jumper selectable voltages of +5.0 VDC and +3.3 VDC with accompanying electrical ground connections. In addition, each of the eight I/O lines is protected from ESD and short circuits. The 12-pin JPA Pmod portal captured in **Photo 2** is configured to power a 3.3 volt Pmod device using external power.

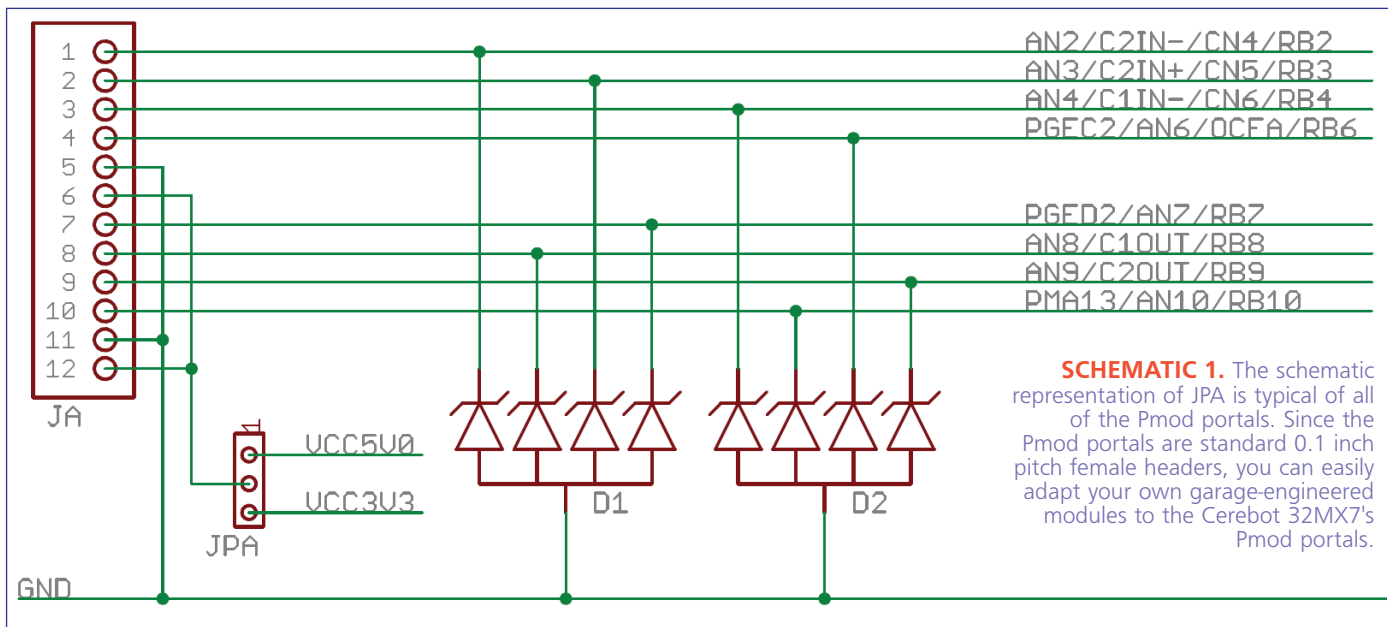
Although Diligent Pmods are design marvels in themselves, there's no reason why your version of a Pmod can't take advantage of the 32MX7's Pmod portals. For instance, **Schematic 1** tells us that Pmod portal JPA exposes a subset of the PIC32's port B I/O pins. Pmod portal JPA also supplies access to the majority of the PIC32's comparator resources and some of the analog-to-



digital inputs. If your personal Pmod needs eight logically consecutive I/O pins, JPB provides access to port E I/O pins RE0 through RE7. As you've already ascertained, each Pmod portal has a differing mix of PIC32MX795F512L I/O. This Pmod portal I/O mix allows you to choose a particular Pmod portal and design your personal Pmod device around the selected Pmod's I/O configuration.

Cerebot 32MX7 USB

The 32MX7's USB circuitry supports USB 2.0 Host, USB 2.0 Device and, OTG (On-The-Go). The Cerebot's USB subsystem can be configured as a USB full speed device or a USB low speed device. USB connectivity is achieved via the PIC32MX795F512L's built-in USB 2.0 engine, a Type A USB connector, and a Mini-AB interface. The 32MX7 can operate as a self-powered USB device or a bus-powered USB device. USB host operation is also possible as the



SCHEMATIC 1. The schematic representation of JPA is typical of a Pmod of the Pmod portals. Since the Pmod portals are standard 0.1 inch pitch female headers, you can easily adapt your own garage-engineered modules to the Cerebot 32MX7's Pmod portals.

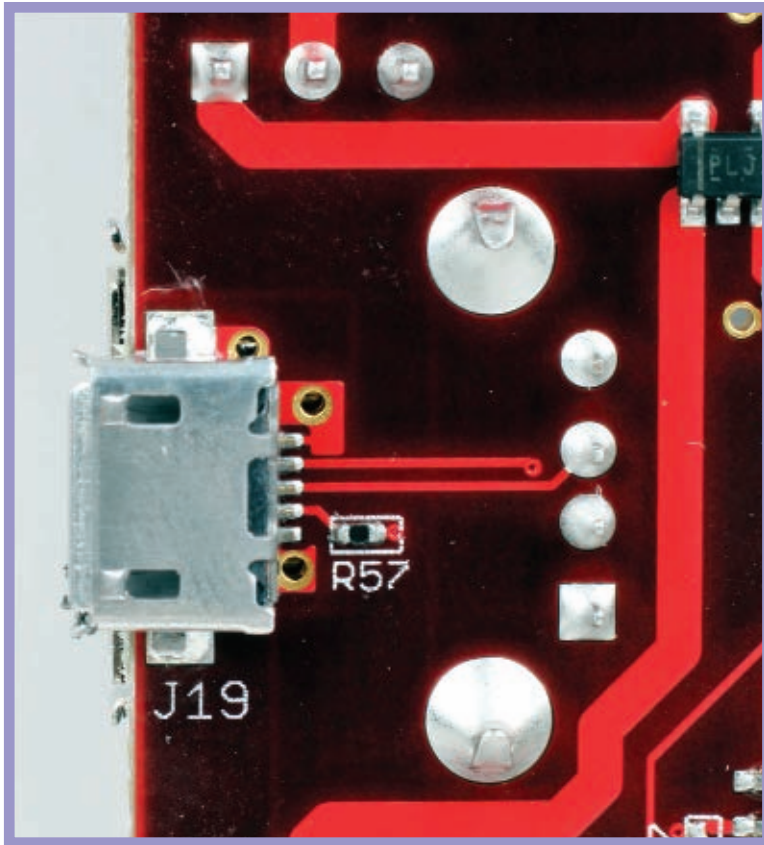


PHOTO 3. This shot shows the Mini-AB USB connector nestled inside of the footprint of the Type A USB connector on the other side of the printed circuit board. This USB connector is normally used for device connections and OTG applications.

connector's footprint. Since I don't know anybody from Krypton, I flipped the Cerebot over and snapped **Photo 3**.

Cerebot 32MX7 10/100 Ethernet

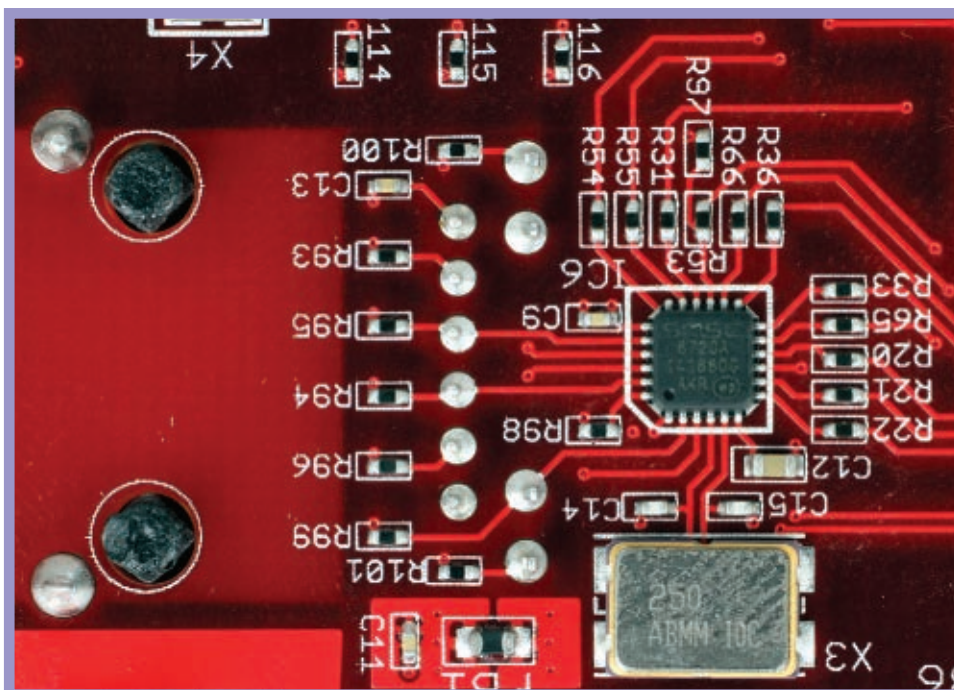
An onboard 10 Mbps/100 Mbps Ethernet interface enables the Cerebot to provide a bidirectional data path for common Internet protocols such as UDP and TCP/IP between the "clouds" and the 32MX7's PIC32MX795F512L. The PIC32's on-chip Ethernet MAC (Medium Access Controller) is paired with an external SMSC LAN8720 Ethernet Physical Layer Transceiver (PHY) to form a wired 10 Mbps/100 Mbps Ethernet portal. A unique hardware address (MAC address) for each 32MX7 is provided via a label on its PCB. Having an official IEEE MAC address enables any number of 32MX7s to interoperate on LANs (Local Area Networks) and the Internet according to the IEEE standards.

The SMSC LAN8720 and its supporting components are shown in **Photo 4**. The role of the PHY component in an Ethernet environment is to interface to the transmission medium. In the case of the LAN8720, the initial point of contact with the transmission medium is the cable plugged into the 32MX7's Ethernet jack. From there, the Cerebot can become part of a LAN, a WAN (Wide Area Network), or ultimately a "cloud." The 32MX7's Ethernet jack is just to the left of the LAN8720 in **Photo 4**.

Cerebot 32MX7 CAN

To prove that CAN networks aren't just for automobiles, take a look at the July '09 issue of *Nuts & Volts*. There, you will find a Design Cycle chronicle in which a bunch of elementary students assembled and utilized a basic CAN network to control a gaggle of small DC motors. As you can see in **Photo 5**,

PHOTO 4. The SMSC LAN8720 is responsible for interfacing the PIC32MX795F512L's internal Ethernet engine to the physical medium. In this case, the physical medium is an Ethernet cable that opens the door to an LAN.



there are a couple of Microchip MCP2551 CAN transceivers mounted on the Cerebot. These two transceivers allow you to set up two separate CAN networks or a single two-node CAN network.

Physically, a CAN network is very simple. A regulation CAN network is terminated with 120Ω resistors at both ends. A Cerebot 32MX7 operating in CAN mode may be in the middle of a CAN device string or at an extreme end of the network. Thus, the Cerebot's CAN termination resistors are jumper selectable. The presence of jumpers JP5 and JP7 in **Photo 6** determine if the associated 120Ω CAN termination resistor is or is not electrically connected to the CAN network.

Cerebot 32MX7 I2C

There are a bunch of really cool I²C devices out there and it would be a shame if we couldn't use them with the Cerebot. So, the designers exposed two of the PIC32MX795F512L's five available I²C interfaces to header points. The first I²C interface (I²C #1) has permanent pull-up resistors and drives the 32MX7's resident 24LC256 EEPROM. I²C #2's pull-ups are actually sourcing current mirrors instead of resistors and can be inserted and removed with the placement of a pair of jumpers. **Photo 7** captures both I²C interfaces with I²C #2's pull-up current mirrors enabled.

Cerebot 32MX7 Power

The Cerebot 32MX7 can be powered externally from the debug USB connector or from the USB device connector. A barrel connector and screw terminal connector act as inlets for external power. The external power source should be a regulated 5.0 VDC supply rated at 500 mA or better. The 500 mA design point is based on the maximum current a USB device can draw from a host. If the current consumption of your Cerebot design will exceed the 500 mA point, Digilent offers an inexpensive 5.0 VDC switching power supply that is rated at 4,000 mA.

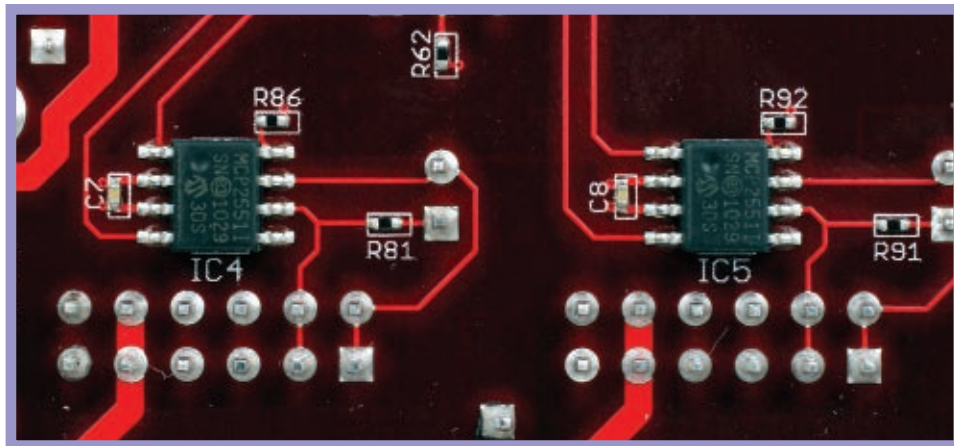


PHOTO 5. Each of the MCP2551 CAN transceivers pictured here is positioned over its respective interface header. R81 and R91 are 120Ω CAN termination resistors. You can see the jumper points that enable termination to the right of each termination resistor.

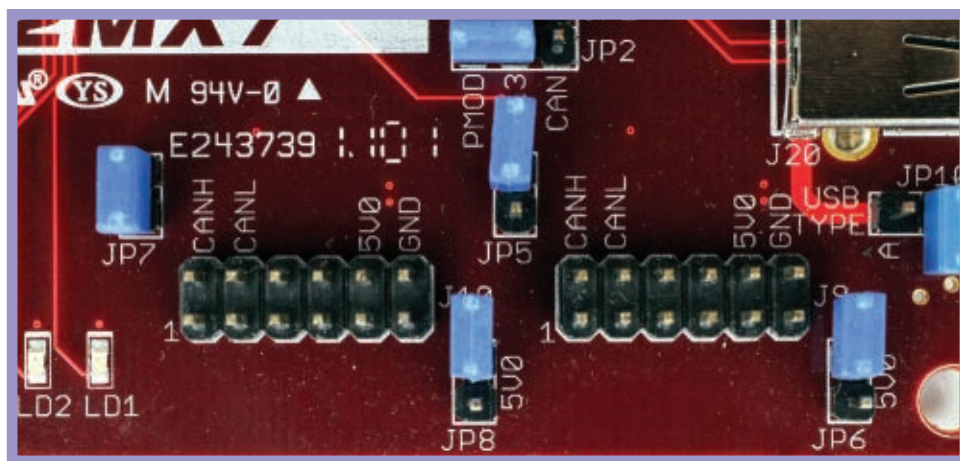


PHOTO 6. Jumpers JP5 and JP7 enable and disable the 120Ω CAN termination resistors. JP6 and JP8 route +5.0 VDC to the CAN headers.

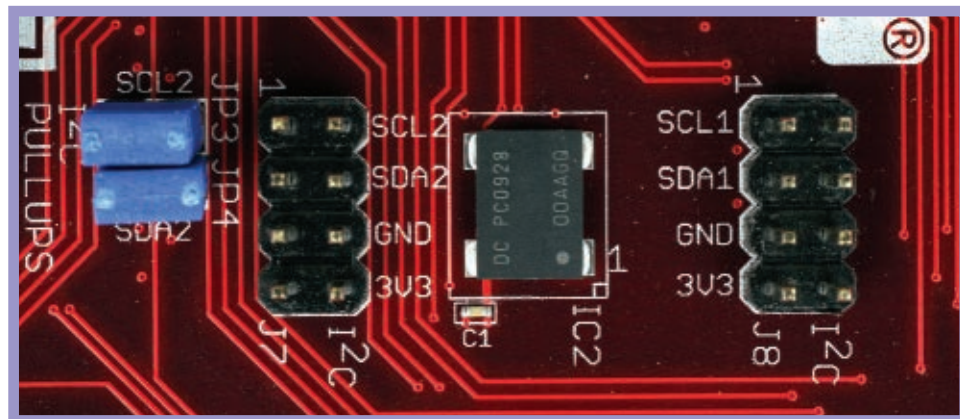


PHOTO 7. The I²C #2 pull-ups are enabled in this shot. IC2 is an 8 MHz resonator that produces the primary clock for the PIC32MX795F512L.

The Cerebot's USB, Ethernet, CAN, and I²C hardware is backed up by free Microchip source code. Drivers and example source code for all of the aforementioned hardware entities can be had for a download from the Microchip website. With that, there's one more Microchip firmware-supported Cerebot communications device we need to examine.

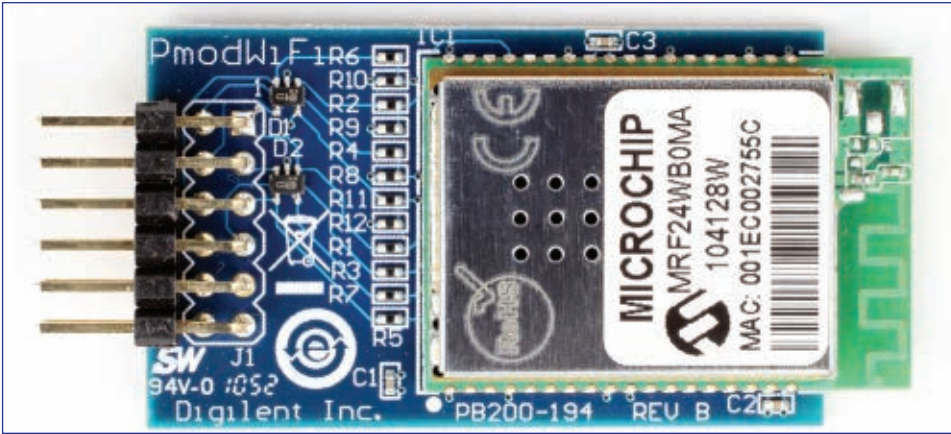


PHOTO 8. Thanks to the Microchip TCP/IP stack, the PmodWiFi is a snap to set up and use. Every respectable robotic unit should be endowed with one of these.

Believe it or not, Pmod portal JPD (which is shown graphically in **Schematic 2**) may be our huckleberry, as Doc Holiday would say. If you're not bilingual as far as SPI dialects go, you won't instantly see that JPD portal is right for the PmodWiFi. A little SPI history lesson

may help.

The SPI protocol was invented by Motorola. True SPI implementations are master/slave concoctions that transfer data in tune with a synchronous clock signal (SCK) emitted by the SPI master node. SPI slaves normally don't speak until spoken to as an SPI slave device is not allowed to generate a data clock signal. An SPI master can clock and exchange data with multiple SPI slave devices.

The data lines to and from the SPI master are shared by all of the slave devices. So, a master can only address and transfer data with one of its SPI slaves at any given time. However, some slave devices have the ability to tap the master on the shoulder when they have data that needs to be transferred.

Data direction is oriented from the SPI master's point of view. Data is transmitted to slave devices on the SPI *Master Out/Slave In* (MOSI) signal line.

Conversely, incoming data from a slave device is driven on the *Master In/Slave Out* (MISO) signal line.

The master selects a slave device to communicate

with by activating the desired slave device's *Slave Select* (SS) signal line.

Okay. Let's perform a Moto-to-Microchip SPI translation against the JPD Pmod portal signal descriptions. Pin 1 of the JPD Pmod portal is labeled SS1 which almost translates directly to the original Motorola Slave Select designation SS. Since there are four PIC32MX795F512L SPI portals, the term SS1 makes sense. Recall that data direction is viewed from the master's point of view.

With that, SDO1 in Motorola SPI speak is equivalent to MOSI. SDI1 is a no brainer and logically translates to MISO. SCK seems to always be the only Motorola SPI term that never needs translation. If you compare the PmodWiFi's pinout in **Figure 1** with the JPD Pmod portal's pinout in **Schematic 2**, you will find that the PmodWiFi is a perfect fit for Pmod portal JPD.

Cerebot 32MX7 PmodWiFi TCP/IP Stack Setup

Now that we have the PmodWiFi hooked up with a 32MX7 Pmod portal, we can begin tweaking the TCP/IP

Connector J1 – SPI Communications		
Pin	Signal	Description
1	~SS	Slave Select
2	MOSI	Master out/Slave in Data
3	MISO	Master in/Slave out Data
4	SCK	Serial Clock
5	GND	Power Supply Ground
6	VCC	Power Supply (3.3V)
7	~INT	Interrupt Output
8	~RST	Hardware Reset
9	~WP	Write Protect
10	HIB	Hibernate
11	GND	Power Supply Ground
12	VCC	Power Supply (3.3V)

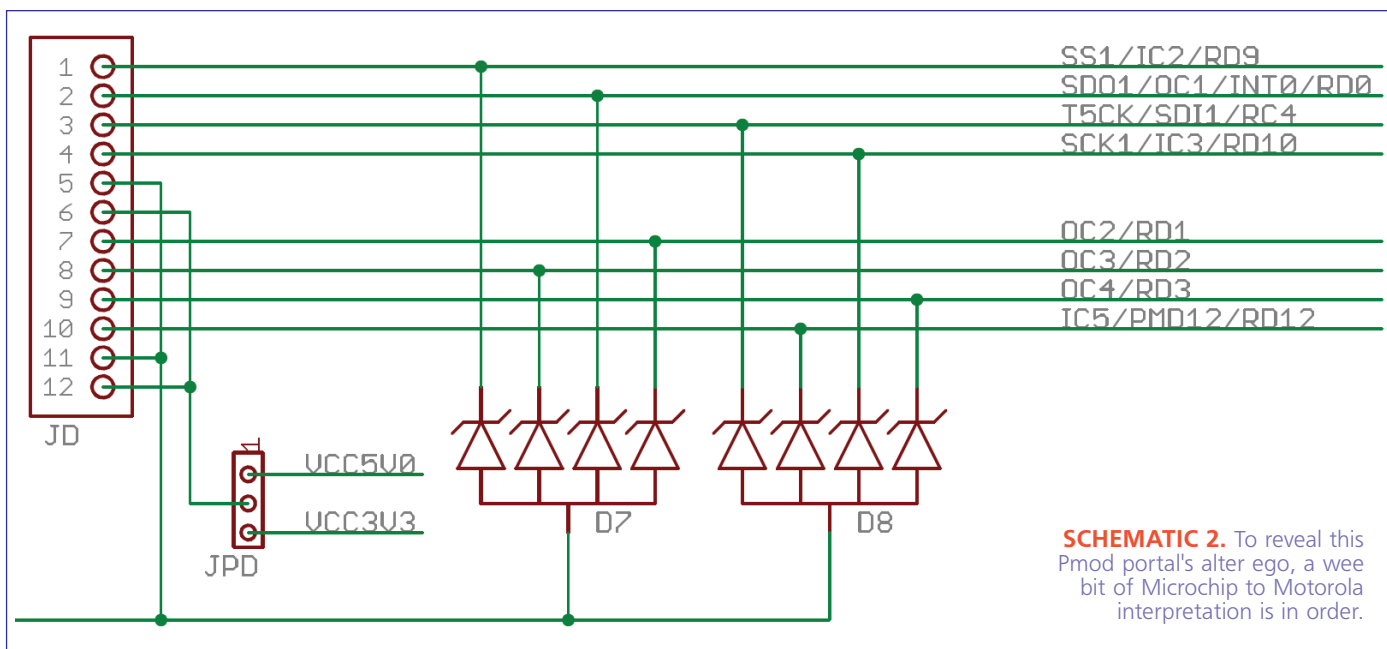
FIGURE 1. The object is to find a Cerebot 32MX7 Pmod portal that can deliver the signals laid out in this PmodWiFi pinout listing.

What You Get

So, what do you get when you cross a Pmod portal with a Microchip MRF24WB0MA? A real live PmodWiFi module like the one that is grinning at you in **Photo 8**. The PmodWiFi is easy to understand, easy to set up, and easy to use thanks to the Microchip TCP/IP stack. With that, let's install, configure, and ultimately use the PmodWiFi.

Obviously, the PmodWiFi is designed to occupy one of the Cerebot's Pmod portals. One indicator is the standard 12-pin Pmod male header. The second clue is the "Pmod" in PmodWiFi. Now that we have discerned where to mount the PmodWiFi, let's see if we can choose the correct 32MX7 Pmod portal.

The logical place to begin is at the PmodWiFi's Pmod connector. The MRF24WB0MA interfaces to a host microcontroller using the SPI protocol. This is evident in **Figure 1** as all of the common SPI signals are present in the PmodWiFi's physical pinout. Now, all we have to do is scan the Cerebot's Pmod portals and select the portal that can most closely support the PmodWiFi's requirements.



stack. The idea is to match up logical I/O definitions in the TCP/IP stack with the physical I/O of the 32MX7. Let's start with mapping the Cerebot's four user LEDs which are physically attached to RG12 through RG15:

```
// TCP/IP Stack Original LED Configuration
#define LED0_TRIS (TRISAbits.TRISA0)
// Ref D3
#define LED0_IO (LATAbits.LATA0)
#define LED1_TRIS (TRISAbits.TRISA1)
// Ref D4
#define LED1_IO (LATAbits.LATA1)
#define LED2_TRIS (TRISAbits.TRISA2)
// Ref D5
#define LED2_IO (LATAbits.LATA2)
#define LED3_TRIS (TRISAbits.TRISA3)
// Ref D6
#define LED3_IO (LATAbits.LATA3)
#define LED_GET() (*(volatile unsigned char*)(&LATA))
#define LED_PUT(a) (*(volatile unsigned char*)(&LATA)) = (a)

// Cerebot 32MX7 LED Configuration
#define LED0_TRIS (TRISGbits.TRISG12)
// Ref D3
#define LED0_IO (LATGbits.LATG12)
#define LED1_TRIS (TRISGbits.TRISG13)
// Ref D4
#define LED1_IO (LATGbits.LATG13)
#define LED2_TRIS (TRISGbits.TRISG14)
// Ref D5
#define LED2_IO (LATGbits.LATG14)
#define LED3_TRIS (TRISGbits.TRISG15)
// Ref D6
#define LED3_IO (LATGbits.LATG15)
#define LED_GET() (*(volatile unsigned char*)(&LATG))
#define LED_PUT(a) (*(volatile unsigned char*)(&LATG)) = (a)
```

I think you get the idea. So, I'll not show you the original TCP/IP stack momentary push button assignments:

```
// Cerebot 32MX7 Momentary Push Buttons
#define BUTTON0_TRIS (TRISGbits.TRISG6)
// Ref S4
#define BUTTON0_IO (PORTGbits.RG6)
```

```
#define BUTTON1_TRIS (TRISGbits.TRISG7)
// Ref S5
#define BUTTON1_IO (PORTGbits.RG7)
#define BUTTON2_TRIS (TRISDbits.TRISD13)
// Ref S6
#define BUTTON2_IO (PORTDbits.RD13)
```

Defining the 32MX7's LEDs and pushbuttons to the TCP/IP stack may seem trivial right now. However, when it gets thick and things aren't working as you would like them to those LEDs and pushbuttons become welcomed debugging tools.

So far, all of the TCP/IP stack redefinitions occur in the HardwareProfile.h file associated with the project. I've chosen to use the TCP/IP stack's 32-bit TCP/IP Demo App, which is based on the Explorer 16 development platform's hardware.

The PmodWiFi SPI I/O assignments are also part of the TCP/IP stack's HardwareProfile.h code. Here are the new PmodWiFi I/O definitions:

```
// MRF24WB0M in SPI1 slot
#define WF_CS_TRIS (TRISDbits.TRISD9)
#define WF_CS_IO (LATDbits.LATD9)
#define WF_SDI_TRIS (TRISDbits.TRISD4)
#define WF_SCK_TRIS (TRISDbits.TRISD10)
#define WF_SDO_TRIS (TRISDbits.TRISD0)
#define WF_RESET_TRIS (TRISDbits.TRISD2)
#define WF_RESET_IO (LATDbits.LATD2)
```

Sources

Digilent
Cerebot 32MX7
PmodWiFi
PmodRS232
www.digilentinc.com

Microchip
Microchip Application Libraries
www.microchip.com

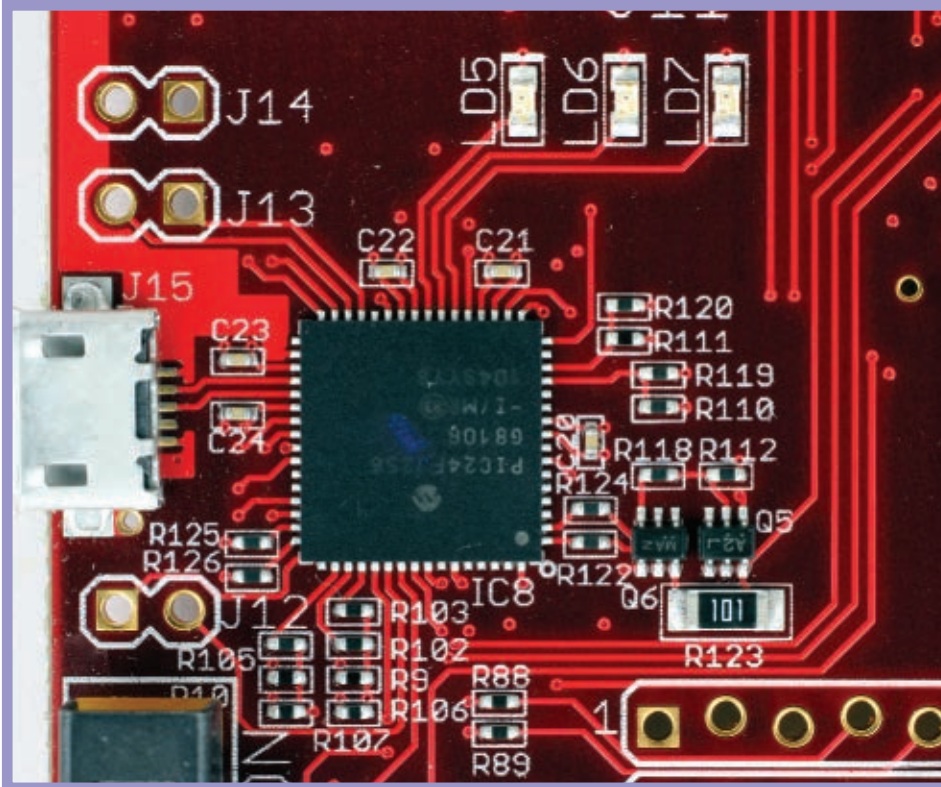


PHOTO 9. Depending on who you talk to, Area 51 either doesn't exist or late 1940's era alien bodies are stored there. The bottom line is that nobody really knows the full story behind Area 51.

and Pmod configurations out of my hat. You can check our work up to this point by downloading the 32MX7 schematic package and PmodWiFi technical information from the Digilent website. The TCP/IP stack and its hardware declarations can be found in the Microchip Application Libraries and is also a download away via the Microchip website. All of these downloads are also available at this article's link.

Our next task involves traversing the main application code and weeding out functionality that we don't want while inserting our desired application attributes. The idea is to initialize the 32MX7

```
#define WF_INT_TRIS      (TRISDbits.TRISD1)
#define WF_INT_IO        (PORTDbits.RD1)
#define WF_HIBERNATE_TRIS (TRISDbits.TRISD12)
#define WF_HIBERNATE_IO  (PORTDbits.RD12)
```

Using SPI #1 is a TCP/IP stack option that we can take advantage of. Since the Cerebot 32MX7's JPD Pmod portal is hardwired to the PIC32MX795F512L's SPI #1 interface, we only have to reassign the PmodWiFi's active-low Interrupt Output (~INT), Hardware Reset (~RST), and active-high Hibernate (HIB) signal lines. The TCP/IP stack does not use the Write Protect (~WP) signal. One potential "gotcha" is that the original ~INT signal line was tied to the PIC32's INT1 pin. If that presents a problem, we'll deal with that when the time comes.

At this point in the Cerebot 32MX7 design cycle, an RS-232 port is a good thing to have. The TCP/IP stack supports serial communications for the purposes of indication and debugging. It's no big deal to configure an RS-232 port in the stack. So, let's check that task off of our to-do list:

```
#define UARTTX_TRIS      (TRISFbits.TRISF5)
#define UARTTX_IO        (PORTFbits.RF5)
#define UARTRX_TRIS      (TRISFbits.TRISF4)
#define UARTRX_IO        (PORTFbits.RF4)
```

As it turns out, the layout of the 32MX7 Pmod portal JPF matches the TCP/IP stack's UART definitions. So, no reconfiguration is necessary. Pmod portal JPF and a PmodRS232 equals instant RS-232 port. With that, we've finished the HardWareProfile.h reconfiguration tasks.

hardware and get the PmodWiFi's radio on the air. Once we accomplish that, we'll configure and install the PIC32 USB device drivers which will give the Cerebot simultaneous Wi-Fi and USB connectivity. We are also going to configure the Cerebot as a USB host and show off its mass storage capabilities.

Area 51

While we're on the subject of Cerebot 32MX7 technical info, take note that there is no schematic representation that explains the hardware shown in **Photo 9**. The word from Microchip is that Area 51 is actually an embedded version of a PICkit3. The only verification of that fact is found in the PICkit3 documentation. The PICkit3 — according to the schematic in the PICkit3 User Guide — is based on a 16-bit PIC24FJ256GB106. Our aerial reconnaissance photo of Area 51 confirms that. I connected a PC to the Area 51 Mini-AB connector and strange flashing lights appeared Northeast of the PIC24. Could this be an alien sanctuary? Or, is it just a cover for some secret Microchip microcontroller project? Microchip and Digilent claim that it is simply a PIC32MX795F512L programmer/debugger complex. (We've heard that cover story before.) Neither Digilent nor Microchip offer any explanation for those mysterious lights.

On a more serious note, in our next Cerebot 32MX7 discussion, we'll finish up the TCP/IP stack install, fire up the PmodWiFi, and load up some USB code. **SV**

What Next??

I didn't pull all of the 32MX7 PIC32 pin assignments

www.servomagazine.com/index.php?/magazine/article/july2011_Eady

Fred Eady can be reached via email at fred@edtp.com.

my minds i.com

BUILD ANYTHING

Ignite your creativity and imagination with the all **new** MINDS-i robotics system. Finally, a rugged **quick-lock** construction system that empowers you to envision, build, and re-create anything that you can imagine with your own "mind's eye".



MINDS-i

Patents US 7,517,270; US 7,410,225 B1; US 7,736,211; US 7,841,923; International Patents Pending.

DO ANYTHING

Utilize PCS Robotics controllers, sensors, and software to bring MINDS-i to life. Use PCS curriculum in the classroom to elevate your student's interest in science, technology, engineering and math!

imagine it



PRO-LITE

explore it



build it



play it



program it



pcsedu.com/servo

I²I Controls

by engineers, for engineers

All SPECTRUM ACE™ CPU boards contain ALEC™. With a built-in text editor, integrated debugging and trace, we make development easy.

SPECTRUM ACE™ and ALEC™ are the perfect combination of power and intellect. Program embedded applications with ease. Our expansion boards all have an ALEC™ command interface, so every environment is fair game.

Start your day with a hot cup of coffee and end it with a hot application!

www.i2icontrols.com

Whether you are a novice or an expert, this is a great way to get started!

The SPECTRUM LITE™ Evaluation Package comes with a SPECTRUM LITE™ CPU, the SPECTRUM LITE™ proto-board, and a 5V DC 1 amp power supply.



\$89.00

CPLDs — Part 5

complex programmable logic devices

A CPLD Project

by David A. Ward

For the final article in this series, I decided to incorporate a CPLD to control a mobile robotics platform or tank-bot (see **Photo 1**). Normally, a microcontroller is used to control the tank-bot, but I thought that it might be fun to see if a CPLD could be used to accomplish some of the same tasks that the microcontroller does. The tank-bot chassis consists of a Tamiya track kit and double gear box mounted on three pieces of aluminum. See **Table 1** for the details and sources for some of the parts used to build the tank-bot chassis.

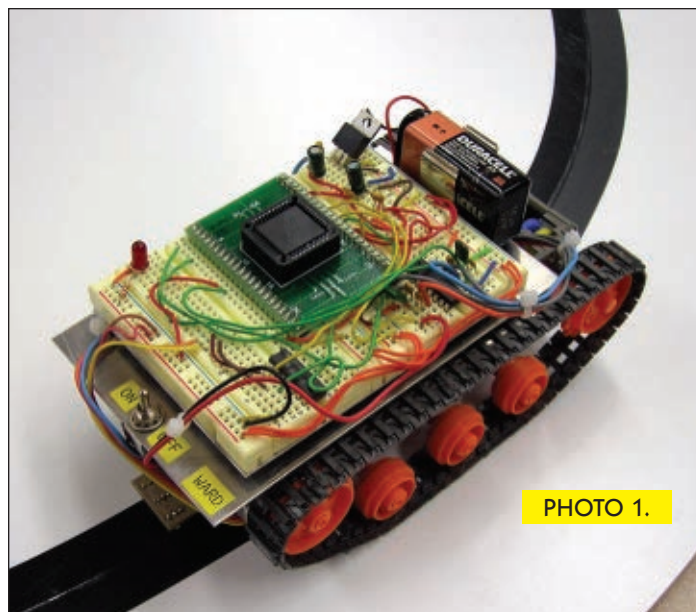


PHOTO 1.

TABLE 1. Tank-bot Parts List.

Track and wheel set	Tamiya #70100	www.pololu.com
Double gear box	Tamiya #70168	www.pololu.com
L293 motor driver IC	L293DNE (296-9518-5-ND)	www.digikey.com
IR LED	106526	www.jameco.com
IR phototransistor	112176	www.jameco.com

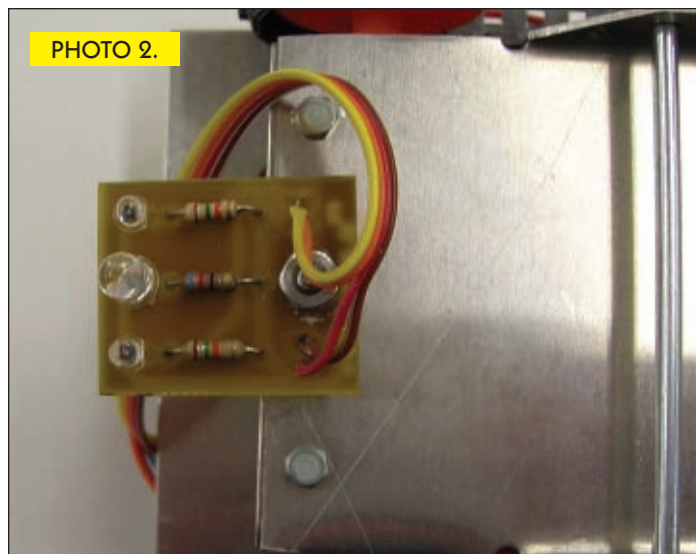


PHOTO 2.

A PCB (printed circuit board) is located on the bottom front of the tank-bot that holds an infrared LED in the center and two infrared phototransistors with one on each side (see **Photo 2**). The LED and phototransistors are used as sensors so the tank-bot can follow a black line of electrical tape laid out on a light colored table top. All of the circuitry involved is shown in the schematic diagram in **Figure 1**. The two phototransistors produce more of an analog output, so Schmitt trigger inverters were placed between their outputs and the CPLD. A L293 quad half H-bridge IC is used to handle the 6 VDC used to energize the two drive motors.

To start a truth table for this project, let's first look at how the infrared phototransistors will behave. When the LED's light reflects off of a light colored object — such as a light colored table top — back to the two phototransistors, they will be in an ON state. When these transistors are ON, they will take their collectors to ground and the CPLD pins high since their signals are being inverted through the Schmitt inverters. This would be the case if the tank-bot were placed on a table top without any black line under the

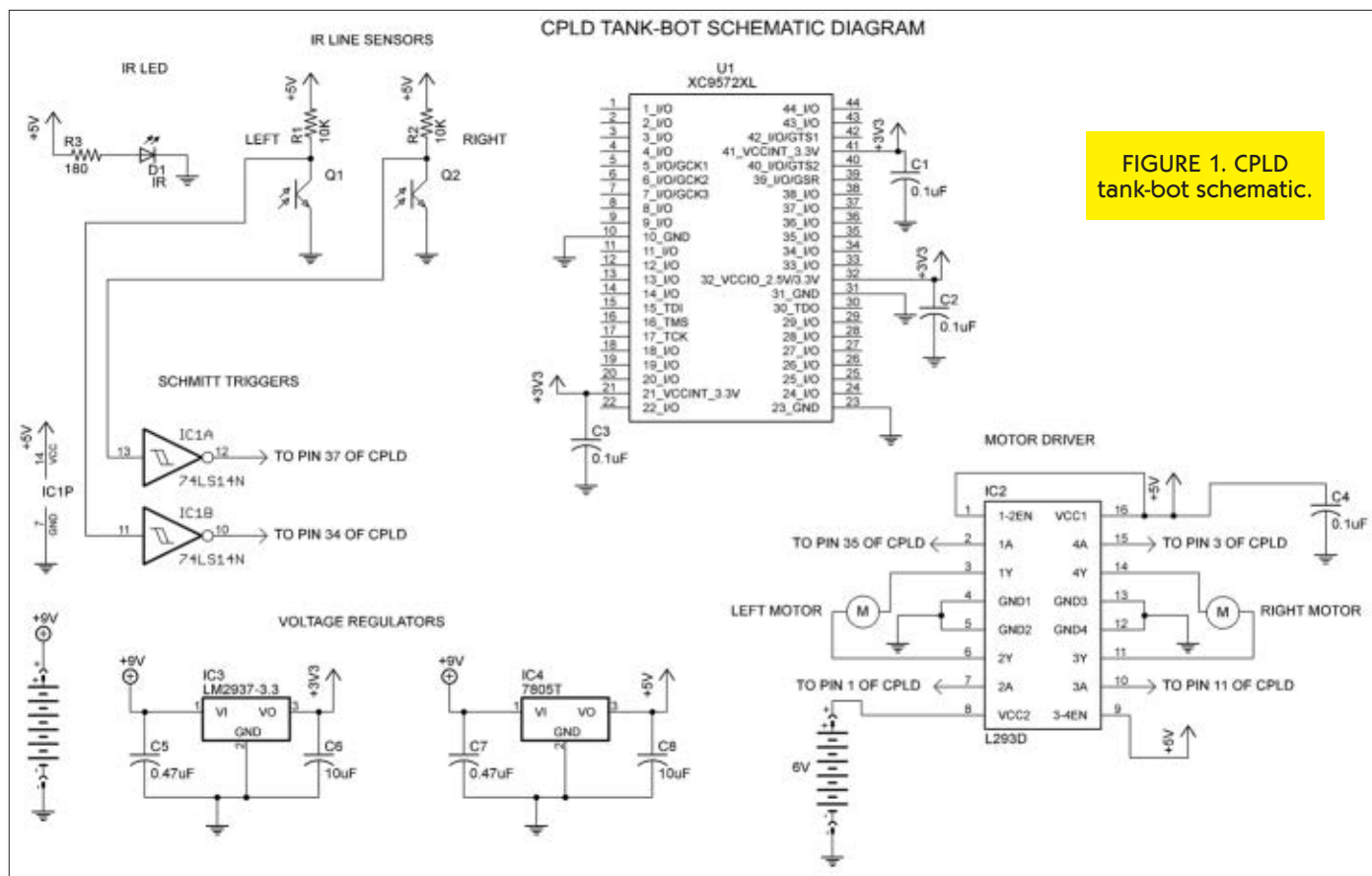


FIGURE 1. CPLD tank-bot schematic.

phototransistors. If we place the tank-bot centered on a black line, then the light from the LED will not be reflected back to the two phototransistors, and they will both be in the OFF state. When they are in the OFF state, the collectors are pulled up to +5 VDC through the 10K resistors. This will make the two CPLD input pins low after they pass through the Schmitt inverters. Now, if the tank-bot is following a black line and it begins to drift to the left of the line, the LED will begin reflecting back into the left phototransistor turning it ON and changing its output from a 1 to a 0. The same will happen to the right phototransistor if the tank-bot drifts to the right. All we need to do to follow the line then is to turn the tank-bot to the right when it's drifting to the left of the line and turn it to the left when it's drifting to the right. The operation of the tank-bot will be as follows:

When placing the tank-bot down on a reflective surface, both motors will be off and the tank-bot will not move. Next, when the tank-bot is placed on a black line the

TABLE 2. Tank-bot CPLD Truth Table.

	Inputs (2)	Photo-transistors (after inversion)	Outputs (4)	Motors			
Condition	Right	Left	L2	L1	R2	R1	Comments
No line present	1	1	0	0	0	0	Both motors stopped
Drifting to right	1	0	0	1	1	0	Right motor forward, left motor reversed
Drifting to left	0	1	1	0	0	1	Left motor forward, right motor reversed
On line	0	0	1	0	1	0	Both motors forward

L293 motor driver connections:

Pin 35 of CPLD (L1) connects to pin 2 (1A) of L293

Pin 1 of CPLD (L2) connects to pin 7 (2A) of L293

Pin 11 of CPLD (R1) connects to pin 10 (3A) of L293

Pin 3 of CPLD (R2) connects to pin 15 (4A) of L293

L293 Truth Table

(Same for both sides 1A, 2A (left motor) and 3A, 4A (right motor):

1A	2A	
0	0	Motor stopped
1	1	Motor stopped
0	1	Motor forward
1	0	Motor reversed


```

32 entity tank_bot_1 is
33     Port ( L : in  STD_LOGIC;
34           R : in  STD_LOGIC;
35           L1 : out STD_LOGIC;
36           L2 : out STD_LOGIC;
37           R1 : out STD_LOGIC;
38           R2 : out STD_LOGIC);
39 end tank_bot_1;
40
41 architecture Behavioral of tank_bot_1 is
42
43 begin
44
45     R1 <= L AND NOT R;
46     R2 <= (NOT L AND R) OR (NOT R AND NOT L);
47     L1 <= NOT L AND R;
48     L2 <= (L AND NOT R) OR (NOT L AND NOT R);
49
50 end Behavioral;

```

FIGURE 2.

```

***** Mapped Logic *****
L1 <= (R AND NOT L);
L2 <= NOT R;
R1 <= (NOT R AND L);
R2 <= NOT L;
Register Legend:
FDCPE (Q,D,C,CLR,PRE);
FTCPE (Q,D,C,CLR,PRE);
LDCP (Q,D,G,CLR,PRE);

```

FIGURE 3.

tank-bot motors will both be put in a forward mode. As the line bends or the tank-bot drifts to the left or right of the line, one of the motors will be reversed to turn the tank-bot back onto the line again. The truth table will have two inputs: one for the left phototransistor (L) and one for the right phototransistor (R). It will have four outputs: two for the left motor (L1 and L2) and two for the right motor (R1 and R2). If the output bits to the L293 motor driver are both a 1 or both a 0, (for example, if 1A and 2A are both a 1 or a 0 at the same time), the left motor will stop because it has the same voltage potential applied to both sides of the motor at the same time. In order to make a motor go forward, you must place a 1 on one input of the L293 and a 0 on the other input — the “A” pins. Reversing the 1s and 0s to the A pins will then reverse the polarity applied to the motors and they will reverse direction. The completed truth

```

66 -- Instantiate the Unit Under Test (UUT)
67 uut: tank_bot_1 PORT MAP (
68     L => L,
69     R => R,
70     L1 => L1,
71     L2 => L2,
72     R1 => R1,
73     R2 => R2
74 );
75
76 L_IN: process
77 begin
78
79     L <= '0'; wait for 1us;
80     L <= '1'; wait for 1us;
81
82 end process;
83
84 R_IN: process
85 begin
86
87     R <= '0'; wait for 2us;
88     R <= '1'; wait for 2us;
89
90 end process;

```

FIGURE 4.

table is shown in **Table 2**. We will use HDL to enter the logic needed into the Xilinx program. Four equations will be generated — one for each motor output. The equation for R1 is: L AND NOT R. The equation for R2 is: (NOT L and R) OR (NOT R AND NOT L). The equation for L1 is: NOT L AND R. The equation for L2 is: (L AND NOT R) OR (NOT L AND NOT R). No Boolean simplification was performed on any of these equations before entering them into the program. The Xilinx software, however, will simplify equations that are entered into it automatically.

Figure 2 shows the HDL listing for the tank-bot line following routine and **Figure 3** shows the simplified equations that were programmed into the chip. **Figure 4** shows the HDL test bench that was used to simulate the logic. **Figure 5** shows the simulated waveforms that were generated in Isim. The program works well when using a XC9536 CPLD which is a 5 VDC powered chip. It does not work as consistently using the 3.3 VDC XC9572XL chip.

Hopefully, this series of articles will help you get started experimenting with CPLDs. Of course, there is a lot more information and practice needed to become proficient in HDL programming but at least you have a starting place and can enjoy a little success early on. **SV**



FIGURE 5.

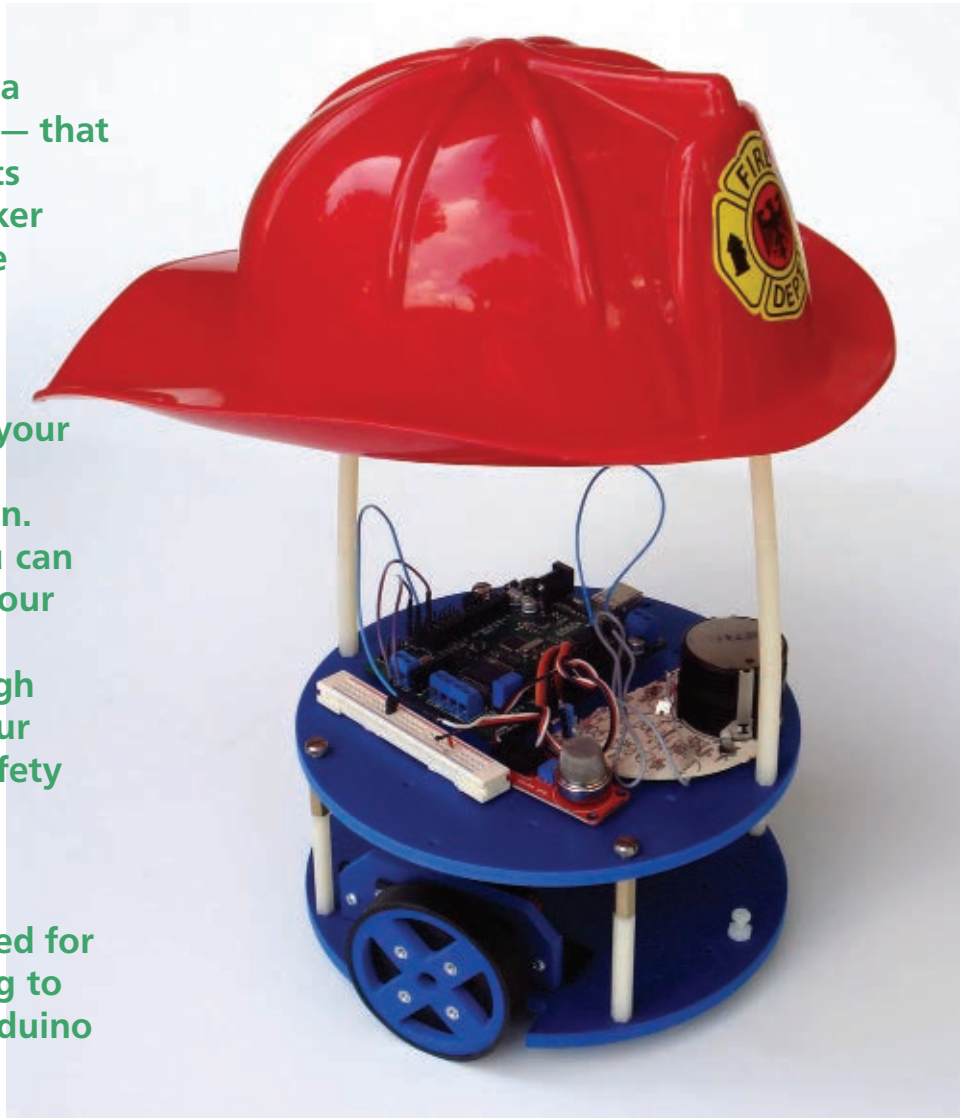
Adding Smoke, Gas, and Heat Detection to Your Robot

by Gordon McComb

Everyone complains that a robot is good for nothin' — that is, other than providing its master with a way to tinker with gadgets in the name of "science!"

However, here's a useful application you can give your robot in short order: fire, smoke, and heat detection. As this article shows, you can easily attach sensors to your robot to detect smoke, poisonous gasses, and high temperatures, making your robot a kind of mobile safety inspector.

Let's take a look at three hands-on projects designed for quick and easy interfacing to any robot built on the Arduino development platform.



Smoke Detection

Where there's smoke, there's fire. Statistics show that the majority of fire deaths each year are caused not by burns, but by smoke inhalation. For less than \$15, you can add smoke detection to your robot's long list of capabilities, and with a little bit of programming have it wander through the house checking each room for trouble. You'll

probably want to keep it in the most "fire-prone" rooms, such as the basement, kitchen, laundry room, and robot lab.

You can build your own smoke detector using individually purchased components, but some items — such as the smoke detector cell — can be hard to find. Instead, you can use a commercially available smoke detector and

modify it for use with your robot. In fact, the process is so simple that you can add one to each of your robots. Tear the smoke detector apart and strip it down to the base circuit board.

Many smoke detectors use an ionization chamber with a mildly radioactive substance: Americium 241. This human-made element has a half-life of several hundred years, but the ionization chamber can become ineffective after less than 10 years due to dust and other contaminants. The same is true of smoke detectors that rely on photoelectric sensors to monitor the amount of smoke particles in the air. Using a very old surplus or used alarm will render your "Smokey the Robot" fairly useless at detecting the smoke of fires, so be sure to tear apart a new detector.

Hacking a Smoke Alarm

You can buy quality smoke alarms for as little as \$7 to \$10. I'll discuss hacking a commercial smoke alarm — specifically a First Alert model SA300 — so it can be directly connected to a robot's computer port or microcontroller. Smoke alarms are not all designed the same, but on many the basic construction is similar to what will be described here. You should have relatively little trouble improvising using most any smoke detector you happen to have. To complete the job, you'll need a digital multimeter to locate proper connection points inside the alarm.

However, you should limit your hacking attempts to those smoke alarms that use traditional nine volt batteries. Certain smoke alarm models — particularly older ones — require you to use AC household power or specialized batteries. These are not suitable for use on your battery-powered bot.

Start by checking the alarm for proper operation. If it doesn't have one already, insert a fresh battery into the battery compartment. *Put plugs in your ears* (or cover up the audio transducer hole on the alarm). Press the "Test" button on the alarm; if it is properly functioning, the alarm

should emit a loud piercing tone.

If everything checks out okay, remove the battery and disassemble the alarm. Less expensive models like the SA300 will not have screws but instead will use a "snap-on" construction. Use a small flat head screwdriver to unsnap the snaps.

Getting Inside the Smoke Alarm

Inside the smoke detector is a circuit board that consists of the drive electronics and the smoke detector chamber.

Either mounted on the board or located close by will be the piezo disc used to make the loud tone. Remove the circuit board, being careful not to damage it. Examine the board for obvious "hack points," and note the wiring to the piezo disc. More than likely, there will be either two or three wires going to the disc:

- *Two wires to the piezo disc:* The wires will provide ground and +V power. This design is typical when you are using all-in-one piezo disc buzzers in which the disc itself contains the electronics to produce the signal for audible tones, or the disc is used as a simple speaker.
- *Three wires to the piezo disc:* The wires will provide ground, +V power, and a signal that causes the disc to oscillate with an audible tone.

Find the wire that serves as ground. On the SA300, it's easy because the battery terminals are labeled + and – right on the board. Find the ground (negative or –) terminal and connect the COM black lead from your multimeter to it. Connect the red test lead to one of the wires or connections to the piezo disc.

Replace the battery in the battery compartment, and depress the "Test" button on the alarm. Watch for a change in voltage. For a two-wire disc, you should see the voltage change as the tone is produced. For a three-wire disc, try each wire to determine which produces the higher voltage; that is the one you want to use. If you are using an oscilloscope, find the wire that produces the cleanest on/off pulse.

Once you have determined the functions of the wires to the piezo disc, remove the disc and save it for some other project. Retest the



FIGURE 1. A First Alert SA300 smoke alarm, before disassembly and after. The hacked alarm has its piezo transducer removed, and a connection wire soldered in its place. See the text for information on how to locate the connection point.

alarm's circuit board to make sure you can still read the voltage changes with your multimeter. Then, clip off the wires to the battery compartment, noting their polarity.

On the First Alert SA300, the piezo disc has three leads; the one that produced the largest voltage change was in the upper right corner. So, I soldered a long "tap-off" wire to it from the underside of the printed circuit board (PCB). On an oscilloscope, this voltage change showed as a series of very fast saw tooth pulses; these pulses are what actually produce the high-pitched tone on the piezo disc. **Figure 1** shows the SA300 unit, before disassembly and once gutted. The hacked board shows the piezo disc removed and tap-off wire soldered to the PCB.

Interfacing the Alarm to an Arduino

Figure 2 shows how to interface the output of a battery-operated smoke detector to an Arduino operating at five volts. Though the Arduino has modest over-voltage protection on its inputs, to be on the safe side the circuit uses a 5.1 volt zener diode and current limiting resistor to help prevent trouble. Most battery-operated smoke alarms operate at up to nine volts (a few to 12 volts), and it's a good idea to limit the voltage going to the Arduino to no more than five volts.

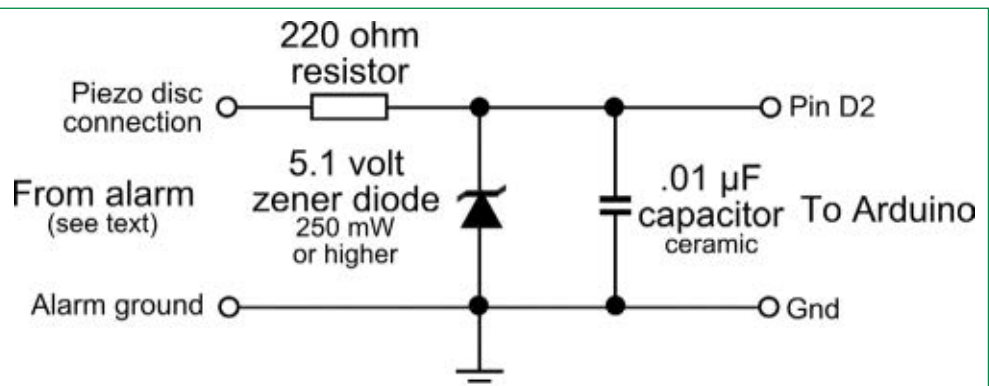
Warning! There's always a chance of damaging your microcontroller when connecting it to alien devices. Exercise care and *proceed at your own risk!*

The circuit in **Figure 2** assumes an output voltage to the piezo sounder of the smoke detector of 7-12 volts. You may wish to add an even more robust buffer circuit for added protection. If something goes wrong — which is always possible when hacking things — it'll be the buffer that needs to be replaced, and not the whole microcontroller!

For example, an opto-isolator provides protection between the smoke alarm and the Arduino. The opto-isolator bridges the gap between the detector and the rest of the robot's electronics.

If you don't feel comfortable with the risks involved, you should opt for a specialty smoke detector module which provides a known output voltage. (These aren't quite as sensitive to smoke as detectors using ionization and photoelectric cells, but they do a

FIGURE 2. A 5.1 volt zener clamps the output of the smoke alarm to about five volts, helping to protect the input of the Arduino (which already has some level of input protection). You may also wish to add extra protection in the form of an opto-isolator.



Listing 1

```
// Smoke detector test
// Detector to digital pin D2
const int alarmPin = 2;
const int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(alarmPin, INPUT);
}

void loop(){
  if (digitalRead(alarmPin) == HIGH) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }
}
```

good enough job for use on a robot.) An example is the MQ-2 flammable gas and smoke detector module, sold by Pololu (item #1480) and others. It's a close cousin to the MQ-5 sensor described later in this article.

By way of example, let's assume that the Arduino periodically polls the input pin that is connected to the smoke alarm circuit board. The program (shown in **Listing 1**) checks the smoke alarm input on pin D2 several times each second. When the pin goes HIGH, the smoke alarm has been triggered, and the Arduino's LED lights up.

If your microcontroller supports hardware interrupts — which is the case with the Arduino — a better scheme is to connect the smoke alarm circuit board to an interrupt pin. Then, write your software so that if the interrupt pin is triggered, a special "I smell smoke" routine is run. For more information on this technique, see the *attachInterrupt* statement on the Arduino Language Reference page at <http://arduino.cc/en/Reference/HomePage>.

The benefit of an interrupt over polling is that the latter requires your program to constantly branch off to check the condition of the input pin. With an interrupt, your software program can effectively be ignorant of any smoke detector functionality. If and when the interrupt is triggered because the smoke alarm circuit was tripped, a special software routine takes over, commanding the robot to do something else. Once the smoke alarm circuit board is connected to the microcontroller, test it and the software by triggering



FIGURE 3. This “air quality” sensor detects propane, LPG, and other toxic and flammable gasses. It needs only a rudimentary interface circuit, and connects directly to an analog input. Photo courtesy Parallax, Inc.

the “Test” button on the smoke alarm. The software should branch off to its “I smell smoke” subroutine. For a final test, light a match, and then blow it out. Wave the smoldering match near the smoke detector chamber. Again, when the detector samples the smoke particles in the air, the output changes and the software runs the “I smell smoke” subroutine.

Running the Smoke Alarm on Five Volts

I found the SA300 unit operated under five volts — that

is, it triggered and sounded its alarm when using the Test button. I have no idea if the unit’s smoke detection abilities are diminished at the reduced voltage, but I suspect this may be the case. Anyway, running the board at five volts saves you from having to power it from a separate battery, so it’s worth looking into.

On the other hand, at the reduced voltage there wasn’t enough rise in signal to trigger the Arduino; the voltage at pin D2 wasn’t enough to register as a HIGH. This occurred whether or not the zener diode was used. An option in this case is to connect a voltage comparator to the hacked output of the smoke detector, and adjust its reference voltage to trigger at the reduced signal level (in my tests, it was a little under two volts).

Limitations of Robots Detecting Smoke

You should be aware of certain limitations inherent in robot fire detectors. In the early stages of a fire, smoke tends to cling to the ceilings. That’s why manufacturers recommend that you place smoke detectors on the ceiling rather than on the wall. Only when the fire gets going and smoke builds up, does it start to fill up the rest of the room.

Your robot is probably a rather short creature, and it might not detect smoke that confines itself only to the ceiling. This is not to say that the smoke detector mounted

on even a one foot high robot won’t detect the smoke from a small fire; just don’t count on it. Back up the robot smoke sensor with conventionally mounted smoke detection units, and *do not* rely only on the robot’s smoke alarm.

Detecting Dangerous Gas

Smoke alarms detect the smoke from fires but not noxious fumes. Some fires emit very little smoke but plenty of toxic gasses, and these are left undetected by the traditional smoke alarm. Moreover, potentially deadly fumes can be produced in the absence of a fire. For example, a malfunctioning gas heater can generate poisonous carbon monoxide gas. This colorless, odorless gas can cause dizziness, headaches, sleepiness, and — if the concentration is high enough — eventually death.

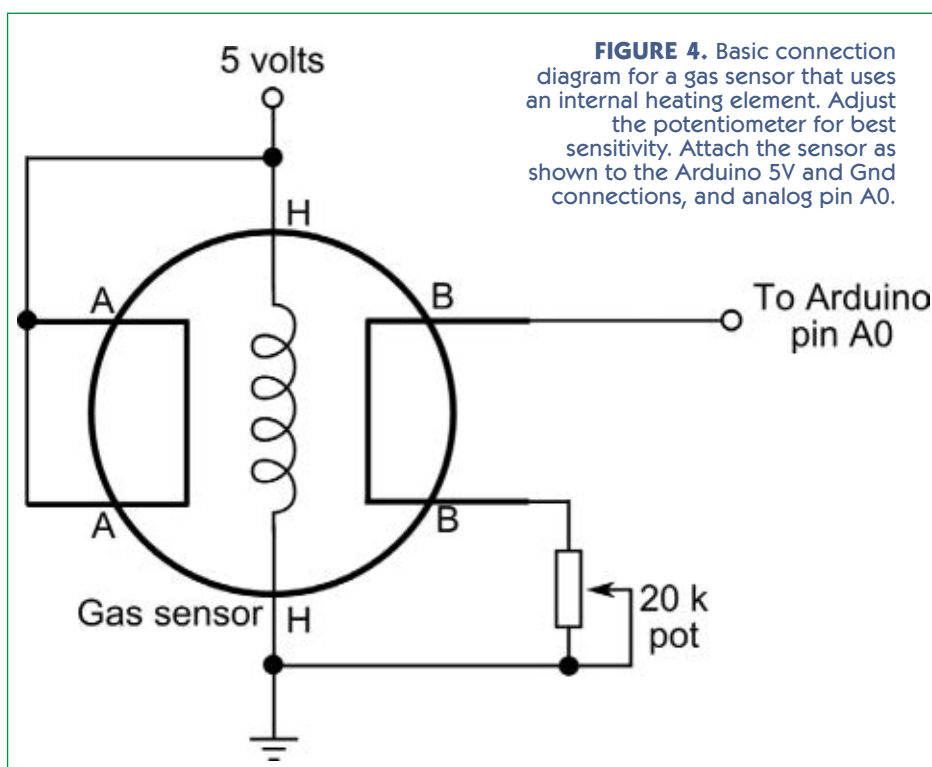


FIGURE 5. Circuit diagram for an LM34 Fahrenheit temperature sensor. For best results, mount the sensor on a small piece of aluminum or other metal, using a thermally conductive adhesive.

Just as there are alarms for detecting smoke, there are alarms for detecting noxious fumes, including carbon monoxide. Such gas alarms tend to be a little more expensive than smoke alarms, but they can be hacked in much the same way. Deduce the signal wires to the piezo disc and connect them (via a buffer such as the resistor and zener diode voltage clamp) to your Arduino.

Combination units that include both a smoke and gas alarm are also available. You should determine if the all-in-one design will be useful for you. In some combination smoke-gas alarm units, there is no simple way to know which has been detected. Ideally, you'll want your robot to determine the nature of the alarm — either smoke or gas (or perhaps both).

Fortunately, there are numerous low cost "air quality" (i.e., gas detection) modules that save you from hacking. The modules are often less expensive than consumer alarms that detect poisonous gas. They're available through Parallax, Pololu, SparkFun, RobotShop, Seeedstudio, and other online retailers.

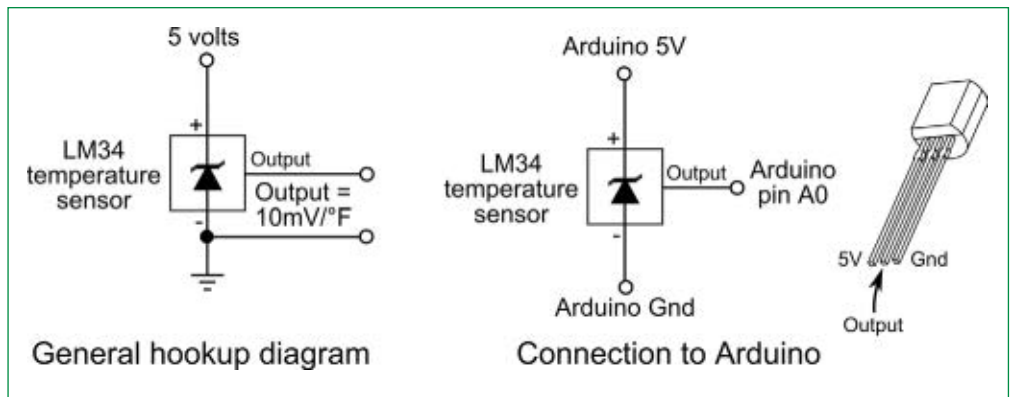
Building a Noxious Gas Detector Circuit

Important! Follow the manufacturer's recommended procedure for testing and using this (or any) noxious gas sensor, or serious injury or death could result. Test only outdoors or in well ventilated areas, and away from flame, sparks, or heat sources. *Use for experimental purposes only!*

Figure 3 shows an example propane LPG sensor and module easily connected to a microcontroller. The sensor is a Hanwei MQ-5 gas detector — capable of picking up many kinds of volatile gasses — with a special sensitivity to LPG and natural gas. A typical hookup diagram is shown in **Figure 4**. A sample sketch for the Arduino is provided in **Listing 2**.

Connect the output of the sensor to analog pin A0.

Note that the



circuit and demo program work with many of the gas and air quality sensors made by the same company (but not all; some need a much more sophisticated connection and heat/cool cycle). For example, the MQ-3 alcohol sensor could be used to create a Breathalyzer-bot. Check the datasheet that comes with the sensor you're using for details.

Adjust the 20 k Ω potentiometer for best sensitivity. Since you probably don't have a calibrated gas source to work from, you'll have to simply guess at the best setting. Start with the pot at its mid-point, and try adjusting it in small steps higher or lower.

Heavy Gas, Light Gas

As with smoke detection, where the gas detector is located has great bearing on whether it detects anything. The reason: Gasses either rise or fall in air, depending on their specific gravity. If a gas is lighter than air, it rises — the same concept behind a helium-filled balloon. Let it go and it charges up into the atmosphere.

- Gasses that are lighter than air — such as methane and natural gas — will rise to the ceiling, and may not

Table 1. Specific Gravity of Selected Gasses.

Gas	Specific gravity	What it does in air
Acetylene	0.9	Slowly rises
Alcohol vapor	1.6	Sinks
Benzene	2.7	Sinks
Butane	2.0	Sinks
Carbon dioxide *	1.5	Sinks
Carbon monoxide	0.9	Slowly rises
Natural gas	0.6	Rises
Methane	0.5	Rises
Propane	1.5	Sinks

*Carbon dioxide is not considered a noxious gas unless its concentration is very high, like when you have too many astronauts in a cramped spacecraft like they did on Apollo 13. Nevertheless, carbon dioxide is considered a dangerous gas when it displaces enough oxygen to make the air "poisonous."

Listing 2

```
// Gas detector test
// Detector to analog pin A0
int val = 0;
void setup() {
  Serial.begin(9600);
}

void loop() {
  val = analogRead(A0);
  Serial.println(val, DEC);
  delay(250);
}
```

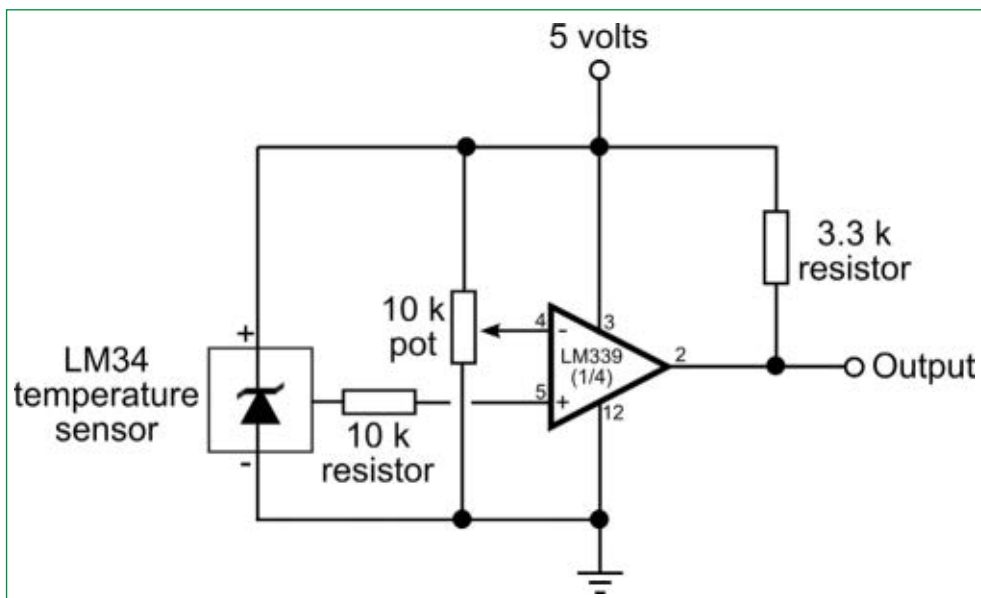



FIGURE 6. Use an LM339 comparator to set a trigger when a certain temperature is reached. The 10 kΩ potentiometer adjusts the temperature at which to trigger the comparator.

— are ideal candidates for robot experimenting.

Sensing Heat

In a fire, smoke and flames are most often encountered before heat, which isn't felt until the fire is going strong. But what about before the fire gets started in the first place, such as when a kerosene heater is inadvertently left on or an iron has been tipped over and is melting the nylon

even be noticed (depending on the concentration) by a gas detector mounted on a robot that's only six inches tall.

- Gasses that are heavier than air or about the same specific gravity, will sink to the ground. Sensors for these gasses — which include benzene and propane

clothes underneath?

If your robot is on wheels (or legs) and is wandering through the house, perhaps it'll be in the right place at the right time and sense these irregular situations. A fire is brewing, and before the house fills with smoke or flames the air gets a little warm. Equipped with a heat sensor, the robot can actually seek out warmer air, and if the air temperature gets too high it can sound an initial alarm.

Realistically, heat sensors provide the least protection against a fire. However, heat sensors are cheap and easy to build. Besides, when the robot isn't sniffing out fires it can be wandering through the house giving it an energy check or reporting on the outside temperature or ... you get the idea.

Figure 5 shows a basic but workable circuit centered around the popular LM34 temperature sensor; there is a general hookup and then one to an Arduino. This device is relatively easy to find online (try SparkFun, RobotShop, and Jameco) and costs just a few dollars. The output of the device (when wired as shown) is a linear voltage. The voltage increases 10 millivolts (mV) for every rise in temperature of 1° Fahrenheit. Program **Listing 3** provides an example for the Arduino microcontroller for reading the sensor.

The LM34 provides output relative to degrees Fahrenheit. Use the LM35 if you want to measure in Centigrade. For more accuracy using either component, you may wish to use a heat-conductive adhesive to glue the flat part of the sensor to a small piece of metal. That will increase the surface area for reading temperature. Heat-conductive adhesives are regularly used to bond microprocessors to heatsinks.

A variation on the LM34 theme is shown in **Figure 6**. Here, the sensor is connected to an LM339 voltage comparator for sensing any temperature above some limit you set via the 10 kΩ potentiometer. For example, you might "calibrate" the circuit to trip at about 100° F. The

Save Up To 60% On Electronic Components

Fantastic New LPC1768 Controller

- * Includes LPC1768 Microcontroller with 512kB Flash Memory
- * 240 x 320 Color LCD with Touch Screen
- * USB, LAN, CAN, SPI, I2C Connections
- * Joy-Stick and SD Card Socket

Only \$119.90

Save On LED and LCD Panel Meters

- * New Digital LED Volt and Amp Panel Meters
- * Easy to Read Ultrabright Displays
- * No Additional Parts Required
- * Ideal for Control Panels and Solar Systems

Starting From Only \$8.90

Great New Touch Keypad

- * Complete and Ready-To-Run
- * Utilizes Capacitive Touch Sensing Technology
- * Runs on both 3.3V and 5V power supplies
- * Outputs both ASCII and Binary Codes for the Key Pressed

Only \$17.90

We are your one-stop shop for Microcontroller Boards, PCB Manufacture and Electronic Components

www.futurlec.com

output of the LM339 comparator will trigger when the temperature exceeds that.

The easiest way to calibrate the circuit is to set the potentiometer when the temperature is what you want. If that's not possible, you can infer the setting by using your digital multimeter to measure the output voltage of the LM34 at some known temperature. Calculate the difference between the known temperature and the desired temperature. Then, multiply this result by 0.01 (remember, the LM34 changes 1 mV per degree F). Add or subtract the difference from the known temperature.

For instance, if the current temperature is 75° and you want to trigger at 100°, the difference is +25°. If the voltage from the LM34 is 2.4 volts at 75°, add 0.25 (25 x 0.01). Set the potentiometer so that the reference voltage to the comparator is 2.65 volts (2.4 + 0.25).

Safety Bot on Patrol!

With the sensors added to your robotic safety guard, you're now ready to program it to sniff out trouble. If your robot is equipped with only rudimentary navigation capabilities, you might have it patrol a single room, stopping for periods of time while it checks its smoke, gas, and heat sensors.

If the bot detects that something is amiss, you can sound an alarm through a piezo speaker; connect the speaker through an amplifier to boost its output if you need more volume.

Sources

Arduino microcontrollers, gas sensors (MQ-2, MQ-5, MQ-6, etc.), and LM34 temperature sensors:

Futurlec
www.futurlec.com

Jameco
www.jameco.com

Parallax
www.parallax.com

Pololu
www.pololu.com

Robot Store
www.robotstore.com

Seed Studio
www.seedstudio.com

SparkFun Electronics
www.sparkfun.com

Gordon McComb may be reached at rbb@robotoid.com.

Listing 3

```
// Temperature sensor test
// Detector to analog pin A0
int lm34 = A0;
int tempF = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  tempF = (500.0 * analogRead(lm34)) / 1024;
  Serial.print("Current temperature: ");
  Serial.println(tempF, DEC);
  delay(500);
}
```

See the *tone* statement on the Arduino Language Reference page for details on how to make sounds through a speaker.

With a robot that can follow walls and navigate around obstacles, you might have it patrol through several rooms. Equipped with an XBee, Bluetooth, or other radio transmitter, you can even beam the current sensor readings back to a PC or other home base. Log the sensor readings at regular intervals to keep a record of the air quality and safety of your home.

So, who said robots are good for nothin'?! **SV**



GREAT FOR
Wireless Data
Links,
Mobile Robots,
Alarm Systems, Sprinkler
Systems & Weather Stations. Anything Remote
that needs a wireless Link. If you have a BASIC
Stamp, now you can program it wirelessly!

FlashFly Serial Wireless Kit \$44.95
FlashFly USB Wireless Kit \$47.95
Program BASIC Stamps Wirelessly,
or use for a wireless data link.
See our site for more info.

USB Stamp Adapter Module \$15.95
Use with FlashFly and program your USB Boe-Bot
wirelessly or it's Great for breadboarding.
(LED PWR indicator and 5.0V regulator built in)

XBee USB Explorer Module \$19.95
Experiment with XBee modules via USB.
(3.3V regulator built in)

10 Pin 2.0mm XBee \$0.95
Female HDR Sockets

BlueWolf ROBOTICS
www.bluewolfinc.com

VISIT OUR ONLINE STORE AT: **WWW.BLUEWOLFINC.COM**
FOR MORE INFORMATION AND OTHER ITEMS.
USE COUPON CODE: **SERVO** FOR 5% DISCOUNT ON ORDERS OVER \$15.00

Adding Vision: Interfacing With RoboRealm

by John Blankenship and Samuel Mishal

For many hobbyists, giving their robot vision is one of their ultimate goals. RoboRealm® is an application for use in computer vision, image analysis, and robotic vision systems. It makes adding vision painless by providing easy-to-use modules that provide color filters, shape recognition, edge detection, and much, much more. This article examines a variety of uncomplicated ways to interface your programs with RoboRealm.

www.servomagazine.com/index.php?/magazine/article/july2011_Blankenship

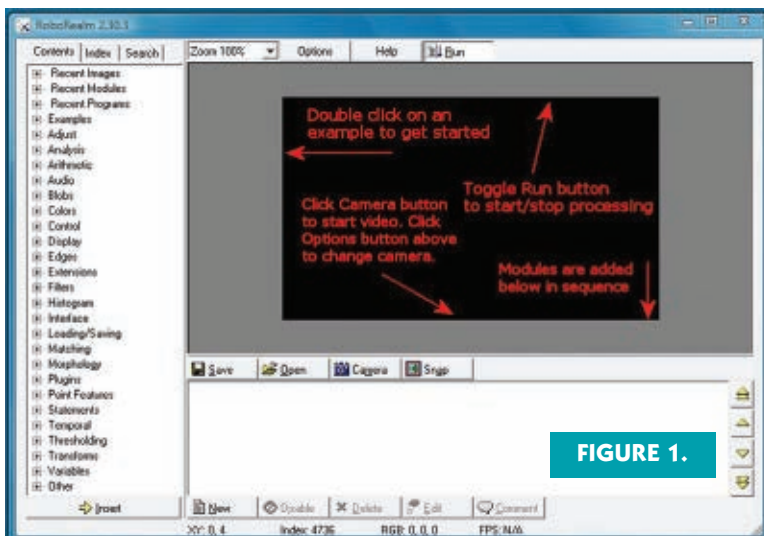


FIGURE 1.

Although vision is a desirable goal for many robot hobbyists, the complex low level programming and mathematical algorithms involved often force many people to assume vision is beyond their capabilities.

Let's examine what is entailed when processing vision. An image to be analyzed can be thought of as three, two-dimensional arrays (one each for red, green, and blue information). The numbers stored in each element of these arrays represent the brightness of a respective x-y coordinate in the image. Processing and analyzing the image involves applying mathematical algorithms to these arrays. Let's look at some examples.

An image can be made brighter by increasing all of the array elements by some percentage. A single color pixel in the image can be converted to black and white by replacing the element in each of the three arrays that is associated with that pixel, with an average of those same three elements. These operations are relatively simple but their complexity can increase quickly as more elaborate actions are needed. Converting a standard image to one showing only the edges of objects, for example, involves finding pixels that differ in color from at least one of their surrounding pixels.

The level of programming and mathematical skills needed to implement sophisticated vision algorithms makes it difficult for many hobbyists to experiment with this subject. Fortunately, a program called RoboRealm provides a solution for many situations by providing a large library of modules that makes it easy to capture images from web cams, and then process and analyze those images. Detailed information — as well as a 30 day trial copy of RoboRealm — can be obtained from **www.RoboRealm.com**.

Although RoboRealm is a programming language itself, it is very different from other languages you might be familiar with. Let's look at some examples to see how RoboRealm can be used to implement a simple vision system. When you first start RoboRealm, you see the opening screen shown in **Figure 1**.

The left side of the screen provides access to the

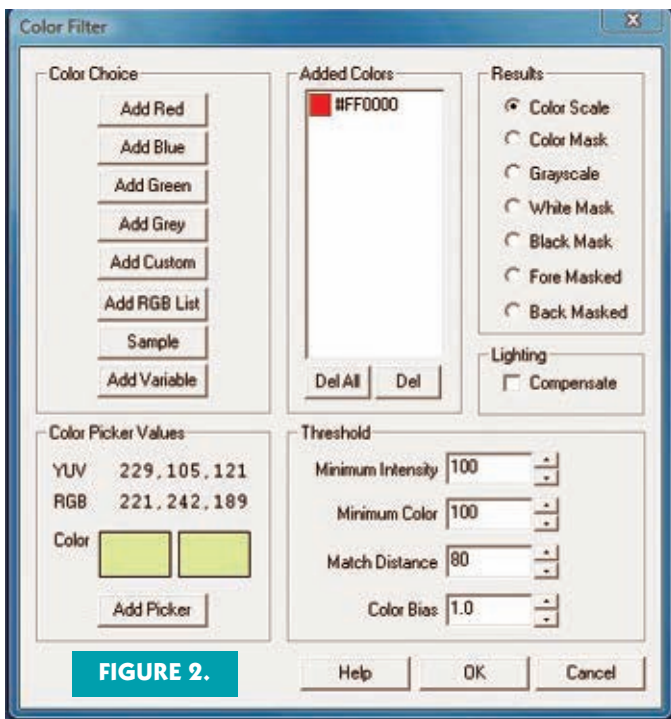


FIGURE 2.

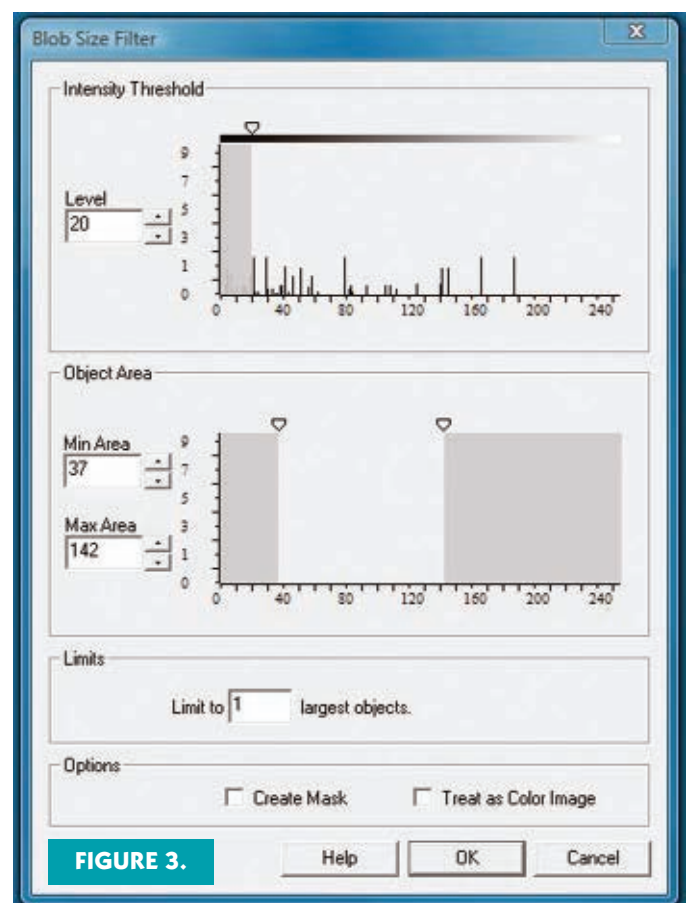


FIGURE 3.

available modules. When a module is selected, it is inserted in the programming window at the bottom of the figure. In general, each time an image is acquired from the active camera, all of the selected modules are applied in consecutive order to that image. Start by clicking the **Options** button and selecting the camera you are using from a drop-down menu. After a camera has been designated, the **Camera** button can then be used to turn it on and off.

In order to demonstrate how easily vision can be implemented, let's create a simple program that can track a red object in the camera's operational view. Expand the **Colors** option in the left column and select **Color_Filter**. When you do, you will get the window shown in Figure 2. You can add basic colors to be detected, as well as choose specific colors from the current image. After you have selected colors (or a variety of shades of the same color) you wish to track, adjust the parameters in the lower right corner of the window so that only the object you wish to track can be seen in the video window of Figure 1.

After your adjustments allow the desired object to be seen as you move it within the camera's view, you are ready to add another module. Expand the **Blobs** option in the left column and select **Blob_Size** to expose the window shown in Figure 3. You can make adjustments so that only groups of pixels (blobs) of a specified size and intensity are displayed. At this point, the video display should show only the movement of the object you wish to track. If necessary, you can fine-tune your options in each of the modules.

Next, expand **Analysis** and select

Center_of_Gravity. You should see the window shown in Figure 4. This module will essentially find the center of a blob of pixels and insert statistics about that blob (such as its **x,y** coordinates) into a number of predefined variables. A box will be drawn around the blob and the coordinates displayed. You can choose to have this information overlaid on the source images as shown in Figure 5.

If everything is adjusted properly, the image will be tagged with coordinates representing its position on the screen when it comes in view. As you move the object, the

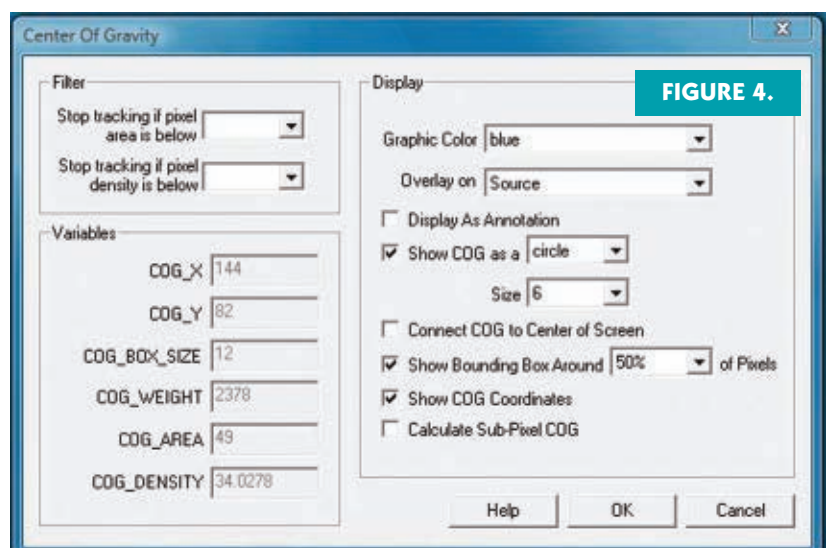


FIGURE 4.

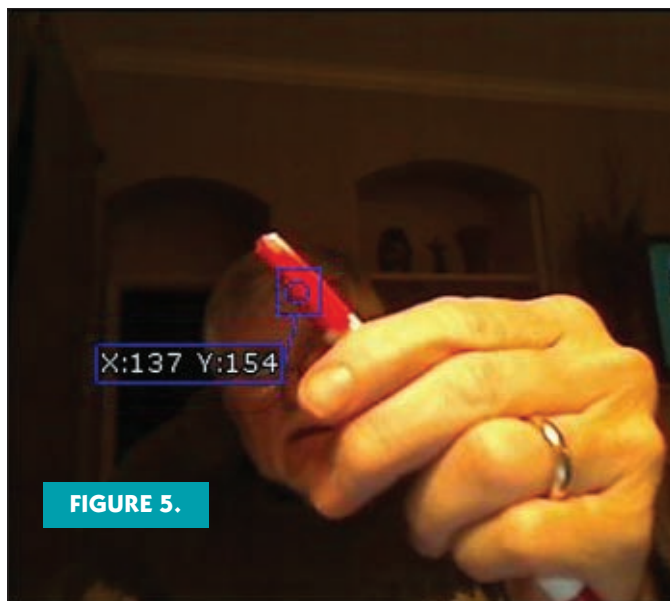


FIGURE 5.

box identifying its position should travel with the object and the coordinates should constantly reflect the current position. **Figure 5** shows RoboRealm tracking the red end

```

Main:
  while true
    call GetCoordinates(a,b)
    xyString 100,100,a;b," "
  wend
end

sub GetCoordinates(&x,&y)
  t=timer()
  x="" \ y=""
  repeat
    getKey k
    if timer()>t+1000 then x=-1\y=-1\return
  until char(k)="x"
  while true
    waitKey k
    if char(k)="y" then break
    if k<>0 then x += char(k)
  wend
  while true
    waitKey k
    if char(k)="z" then break
    if k<>0 then y += char(k)
  wend
  x=ToNumber(x,-1)
  y=ToNumber(y,-1)
  return

```

Figure 6. Extracting the screen coordinates from a keyboard stream.

of a pen. At this point, we have used a series of complex mathematical computations to implement a fairly robust vision algorithm without having to understand or program any of the mathematics. Even though this is a very usable algorithm, it is important to realize that the capabilities of RoboRealm far exceed this simple example. Properly utilizing some of RoboRealm's more complex modules, for example, should allow you to identify and track shapes instead of colors, and even perform facial recognition.

The RoboRealm library is so extensive that a new user can easily be intimidated. However, with experimentation and exploration of the RoboRealm examples, help files, and forum posts, a typical hobbyist should be able to implement vision algorithms that exceed anything they could have done on their own. RoboRealm also has the ability to directly control motors, so a complete robot can be created using RoboRealm alone. Many applications need the features of a specific language though, so it would be useful if the information generated by RoboRealm could be transferred to other programming languages.

Fortunately, RobotRealm has provided many ways to allow the information it generates to be shared with other programs. For example, the data could be sent over a serial link to a microcontroller. While this is an excellent option, it is doubtful that a microcontroller could fully exploit RoboRealm's capabilities. In order for a robot to fully utilize vision, it must be controlled by a full-featured, high level language capable of complex mathematical operations, large multi-dimensional arrays, and many other features not found in the microcontrollers typically used to control hobby robots.

Let's assume an appropriate robot control language is running on the same machine as RoboRealm. Our goal is to demonstrate generic ways of interfacing RoboRealm with the control program so that the vision information can be obtained and utilized to make high level decisions. For our examples, RobotBASIC will be used as the robot control program, but the interfacing methods explored here should work with almost any programming language. RobotBASIC can be downloaded for free from www.RobotBASIC.com.

The first interfacing method we are going to examine is the least sophisticated, but because of its simplicity, it should work with any high level language. RoboRealm provides a **VBscript** module that we use to format the **x,y** coordinates of the blob being tracked into a single ASCII string. To make it easy for the receiving program to extract the data, we formatted the information so that it is sandwiched between the letters **x**, **y**, and **z**. For example, if the blob is located at coordinates 57,109, then the data will be formatted like this: x57y109z.

Once our desired information was formatted, we used RoboRealm's **Keyboard_Send** module to insert the data into the PC's keyboard buffer. This module allows you to specify which application should receive the keystrokes.

The format of the data makes it easy for the receiving program to watch the stream of keystrokes coming from RoboRealm and convert the coordinates to integers. **Figure**

6 shows how easily this can be accomplished with RobotBASIC. The details of the conversion are encapsulated in a subroutine called GetCoordinates. After the routine is called, the two variables in the parameter list will contain the **x,y** coordinates of the blob being tracked.

The subroutine is more complex than necessary because it uses a timer to set the coordinates to (-1,-1) and return control to the main program if data has not been received within 1,000 ms. If the object goes out of view, RoboRealm will send zeros for the x,y coordinates. This means that any values greater than zero can be considered valid.

The logic of this example should be easily implemented in other languages, making it effortless for anyone to experiment with vision. You do not have to understand any of the complexities of the RoboRealm program. Simply download the program and edit the **Keyboard_Send** module so that it sends keystrokes to your program. Note: RoboRealm versions 2.30.6 and later provide greater control over which program receives the keystrokes. You can, for example, select RobotBASIC's terminal output window instead of just RobotBASIC. With earlier versions, you might have to tell RoboRealm to force focus on the intended application.

In order to demonstrate how the information obtained from RoboRealm can be used in a real world situation, we utilized RobotBASIC's graphical capabilities to create a simulation of two eyes being moved by a servomotor, as shown in **Figure 7**. Mount your web cam at the top of your computer screen and move the object to be tracked. As long as you have the appropriate parameters adjusted in RoboRealm, you will see the eyes follow the object. You have to see this program in action to fully appreciate the effect it can have on you.

The code fragment that controls the eyes is shown in **Figure 8**. The full program can be downloaded from the article link, but the listing shows how easily the **x** coordinate can be converted to a pulse width. Of course, you could send the pulse width to a real servomotor controller instead of the simulation.

RoboRealm also has a module that can send data to other applications over the Internet or an LAN. We set up that module for UDP (User Datagram Protocol) communications. This is a viable option for any language that supports UDP sockets. A RobotBASIC program that can capture the UDP data is shown in **Figure 9**. The program starts by printing your computer's IP address so you can use it to set up the RoboRealm **Socket_Communications** module. The translation portion of the program in **Figure 9** differs from the keystroke program in **Figure 6** because the UDP data is captured as a string instead of character by character.

In summary, it is easy to implement powerful vision algorithms in RoboRealm and interface their output to nearly any other program. **SV**

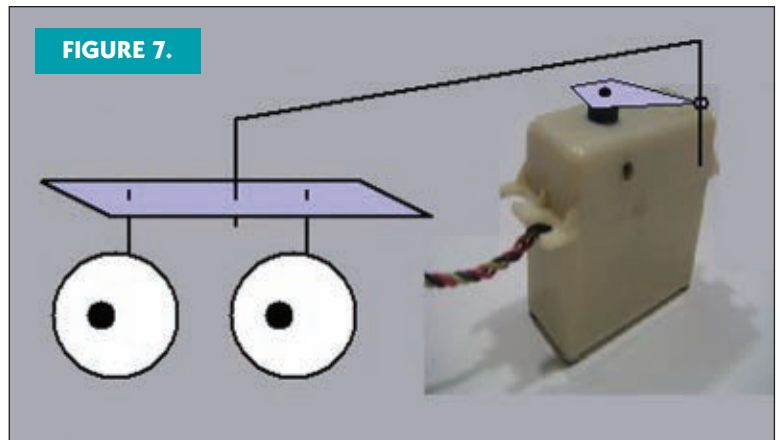


FIGURE 7.

```
while true
  call GetCoordinates(x,y)
  if x>0 and y>0 // valid data
    PulseWidth = 1900-x*2.5
    gosub Servo
  endif
wend
```

Figure 8. Controlling a simulated or real servomotor using the vision data.

```
Main:
  print "Local IP address = ",TCP_LocalIP()
  UDP_Start("MySocket",4040)
  while true
    call GetCoordinates(a,b)
    // print if object is in view
    if a<>-1 then xyString 100,100,a;b,"
  "
  wend
end

sub GetCoordinates(&xCoord,&yCoord)
  xCoord = -1 \ yCoord = -1
  if UDP_BuffCount("MySocket")>0
    d=UDP_Read("MySocket")
    px=InString(d,"x") \
    py=InString(d,"y")
    xCoord = ToNumber(SubString
      (d,px+1,py-px-1),-1)
    yCoord = ToNumber(SubString(d,py+1),-1)
  endif
  return
```

Figure 9. Extracting the screen coordinates from a UDP message.

Twin Tweaks



THIS MONTH: Savox Rocks



SERVOs FROM SAVÖX.

Mastering the fundamentals is critical in any creative endeavor. Aspiring Top Chefs must master a Velouté and the other mother sauces: musicians hoping to hit it big need to start with their chromatic scales. In that same vein, one of the most fundamental hacks for an ambitious roboticist is the classic full rotation servo hack. Servos are a staple in robotics, renowned for their simplicity, affordability, and controlability. However, many RC servos have a limited range of motion because applications like steering the rudder of an RC airplane do not require full

rotation. Many robotics applications, however, do require full rotation. This month, we have the pleasure of presenting a set of servos from Savöx, a company that specializes in high performance servos for RC cars and planes. In addition to testing their stock capabilities, we wanted to execute the classic full rotation hack to see if these servos designed for steering could deliver the same high performance for long distance locomotion.

Quickfire Challenge

THE SAVÖX HORN SET.



Savöx is a company that offers high performance electric motors, servos, and sensors. On the cutting edge of quality with in-house R&D and production, some of Savöx's flagship products are their RC servos. Savöx servos come in a variety of shapes and sizes, and are targeted for use in RC cars, trucks, planes, and boats for things like steering mechanisms. The servos we received were the SC-1258TG model. The technical stats on the servo are impressive. The first thing you notice about it is the sleek case made from aluminum that is not only durable, but also doubles as a heatsink. The SC-1258TG boasts a speed of 0.08 seconds/60° with an impressive torque of 12 kg*cm. This combination is achieved with a gearbox made up of beautifully machined titanium gears. The unit is driven by a small coreless motor which means that the servo has excellent acceleration. The SC-1258TG is a digital servo with 12-bit resolution, ensuring that it is both fast and accurate. Quality comes with a price

of course, and the SC-1258TG will run you \$68 from the Savöx online store but up to \$82 from other distributors. The prices of the various Savöx servos really run the gamut, ranging from above \$100 to below \$30.

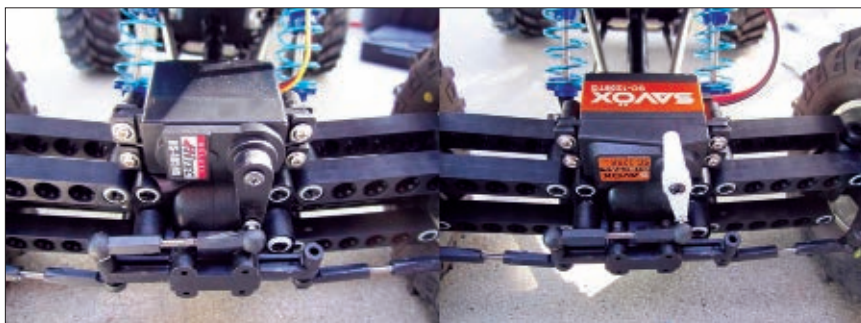
Before embarking upon the full rotation hack, we wanted to get acquainted with the out-of-the-box abilities of the Savöx servos. We didn't have any RC planes to test it in, and we wanted a platform that would take advantage of the servo's excellent torque. It only took the cover of a recent *SERVO Magazine* to provide the perfect inspiration: the Lunar Rover from MINDS-i uses a 180° rotation servo for steering. With big tires and fancy shocks, the Lunar Rover looks the part of an intrepid vehicle that would need strong steering to avoid becoming trapped between a rock and a hard place. The original servo on the Lunar Rover was a Hitec HS485HB – not a shabby servo by any stretch of the imagination.

Attaching the Savöx to the Rover frame was straightforward – the Savöx servos include the classic mounting bracket on the side of the case common to most servos (with the servos from the VEX kit being a conspicuous exception). The Savöx was able to pick up the same mounting screws as the original steering servo, and the final step would be to mate the servo to the steering bar linkage.

One of our major concerns with implementing servos in a robotics kit is how to mechanically mate the servos to the kit. To facilitate that endeavor, the Savöx servos come with a variety of servo horns, ranging from the classic small to disk to a six-legged wheel. We also discovered that servo horns for our Futaba servos fit the 25 tooth spline of the Savöx, but the Hitec servo horns did not. We wanted a horn close to the shape of the one for the original steering servo, and one of our Futaba horn sets had a two armed horn that was just the right length.

The final step was to modify the horn so that it could accommodate the 4-40 attachment screw from the MINDS-i kit. We drilled a pilot hole and then tapped it for a 4-40 screw. The minimalistic horn did not have a lot of excess plastic, but the tapped hole was still flanked by enough material so that it did not compromise the structural integrity of the part. After positioning the horn on the servo and reconnecting the steering linkage, we were ready to test the Lunar Rover's new steering.

The steering did indeed seem more responsive. This makes sense, given that the affordable Hitec servo has a speed of 0.19 seconds/60° and a torque of 6 kg*cm,



REPLACING THE MINDS-I LUNAR ROVER STEERING SERVO.



CONQUERING ROUGH TERRAIN WITH THE SAVÖX.



COMPARING THE GEARBOXES.

meaning that Savöx outclasses it by at least a factor of two in both areas. The improvement was most noticeable when making rapid changes of direction, and in the agility for dodging obstacles like desert tortoises. We even set up an obstacle that we tested with both the original steering servo and the Hitec. Though it was difficult to ever get the energetic Rover stuck anywhere, we were able to inescapably trap it while still equipped with the original steering servo. We found ourselves unable, however, to trap the Rover with the Savöx servo at the helm.

Super steering is awesome, but many robotics kits also



DISASSEMBLING THE SAVÖX GEARBOX.



REMOVING THE PIN.



MESSING WITH THE POTENTIOMETER.

rely on servos to power entire drive trains. Even back when our fledgling FIRST team was experimenting with the EDU-bot, we were working with servos modified for full rotation. For the Savöx servos to be a truly exciting option for roboticists, we had to see if they would be amenable to this classic hack.

Your Rotation has Been Adjusted

Most servos are used for maintaining positions or other limited adjustments, so the default is to have a limited range of motion like 90° or 180°. Servos accomplish this via feedback mechanisms, and in the case of the Savöx and many other servos, this includes a potentiometer and a mechanical stop. The potentiometer is linked to one of the gears in the gearbox, and the changing resistance of the potentiometer allows the servo to know what the angle of rotation is. A physical stop also keeps the servo from making a full rotation. Both of these mechanisms need to be adjusted to allow the servo to experience the joy and freedom of 360° rotation.

The full rotation hack is indeed a classic, and there are numerous resources on the Web that describe the process step by step for a variety of servos. One of our favorite guides is from Acroname Robotics which offers detailed illustrated instructions for modifying the Hitech HS-300. There are many other guides online directed at many other servos, but none for the Savöx servos. The website ServoCity.com even cataloged a rotation modification difficulty list for prospective hackers to let them know what they might be getting themselves into. Some servos have even been deemed unmodifiable. One of the main reasons a servo might not be able to be hacked into full rotation is because some servos only use a half gear in the gearbox. This saves space but also means the gearbox can't support full rotation.

Upon initially opening the Savöx servo, we were afraid that a full rotation hack would be difficult. The titanium and aluminum gears looked positively cozy, and a bearing at the top of the shaft seemed to decree to the gears below that none shall pass. As we fretted over visions of cutting off the bearing with a Dremel tool, we stumbled upon a way to disassemble the gearbox that was far less destructive. Using two screwdrivers and a careful application of force, the gearbox could be pulled apart

enough so that the large gear — bearing intact — could be freed from the gearbox. With this keystone gear of sorts removed, the rest of the gearbox fell away like leaves from the autumn trees.

The easier of the two major modifications that needed to be made was dealing with the mechanical stop. In some

servos, the mechanical stop is a small post protruding from one of the gears — often an actual part of the plastic gear. The small post rides in a groove in the casing, with the arc of the groove circumscribing the arc that the gear is allowed to travel. The Savöx servo used the same type of post common to most servos, but instead of being part and parcel of the gear itself, the stop was actually a removable pin. This was a relief, because cutting off a metal post would not have been pleasant and conjured up images of trying to secure the tiny gear in a tiny vice to cut it with a tiny saw.

That said, we were unsure at first how best to remove the pin because it was a snug press fit. We thought a punch might do the trick, though it would be hard to hold the gear in place, position the punch, and use the hammer all while sparing our fingers. Of course, when you have too many variables, the way to proceed is to remove one from the equation, and removing the finger crushing variable of the manual punch leaves you with an automatic center punch. Serendipitously, our automatic center punch was the perfect diameter to punch out the pin from the Savöx gear.

The second modification involved the potentiometer. Removing the large gear revealed the knob of the potentiometer. To better understand the relationship between the pot and the servo rotation, we wanted to test the effects of adjusting the potentiometer in real time. For that we needed a power source and control for the servo, and the Lunar Rover from MINDS-i once again fit the bill perfectly. We disconnected the steering servo and hooked up the partially dismantled Savöx in its place, using the Rover's radio controller to control the servo. Without the mechanical stop, the servo did indeed rotate freely. Rotating the pot completely in one direction or the other allowed the servo to speed up, but only travel in one direction. Keeping the potentiometer centered allowed the servo to rotate continuously in both directions, but it did not appear



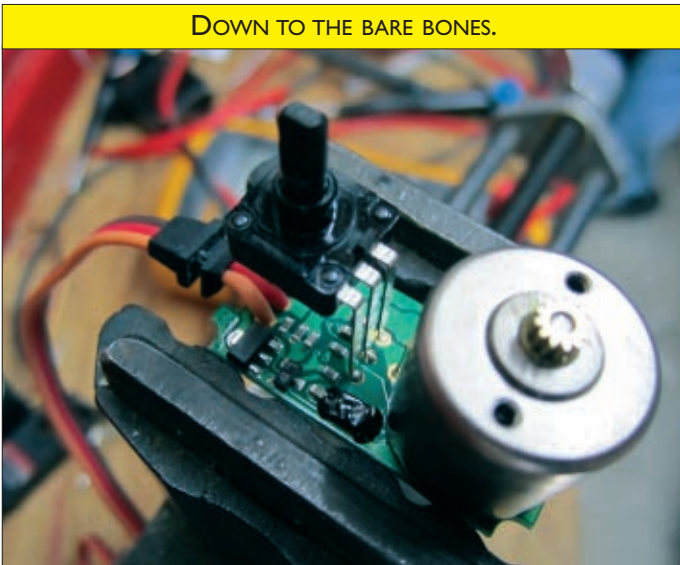
STRUCTURALISM.

to be as fast as it was for unidirectional rotation.

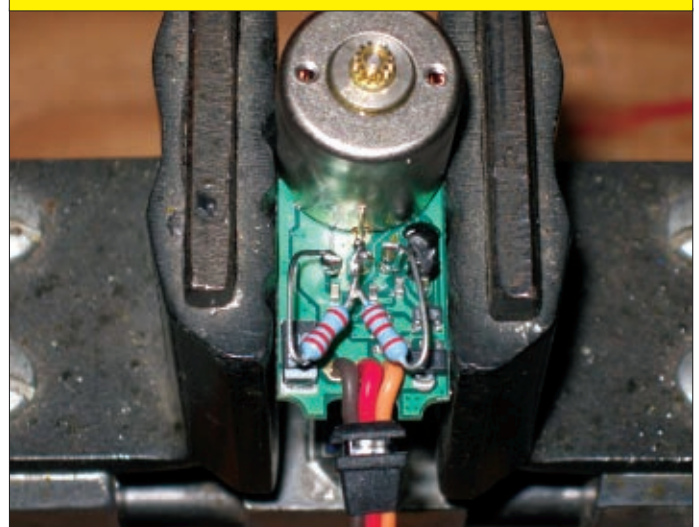
The final disassembly of the servo was a bit tricky because the heads of the final attachment screws were actually inside the casing. Approaching from the bottom of the servo, the PCB (printed circuit board) could be nudged out of the way just enough to allow a skinny screwdriver to reach the final screws. With the casing out of the way, the servo was stripped down to just the PCB, potentiometer, and coreless motor. Because the servo rotation was the fastest in both directions when the potentiometer was set to its center position, the final task was to trick the servo into thinking that the pot was always set to that position.

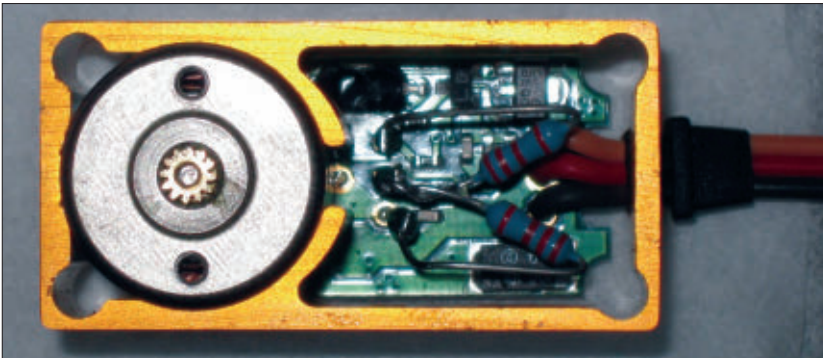
There are two ways to achieve this trickery. The option we pursued was to replace the pot with a small voltage divider circuit. The classy potentiometer in the Savöx servo made this an easy task because it was soldered into the

DOWN TO THE BARE BONES.

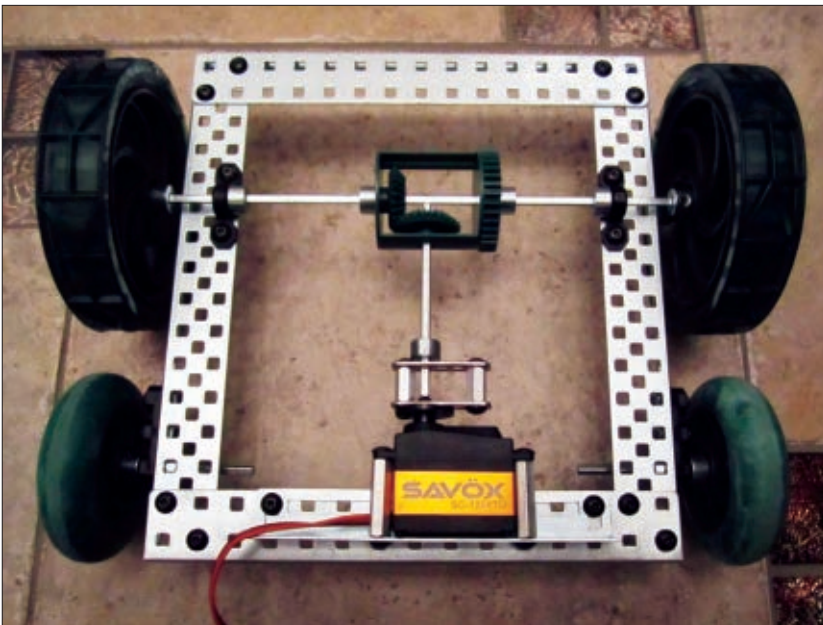


ADDING THE RESISTORS TO REACH FULL ROTATION POTENTIAL.





PUTTING IT ALL BACK TOGETHER.



A DRIVING BASE FOR THE SAVÖX.



FEATS OF STRENGTH AND SPEED.

PCB on long posts instead of wires. To remove the pot, we cut it at the posts, with the posts being the perfect place to mount our replacement voltage divider. Wires would have been more of a bother because if they weren't long enough to resolder to, they may need to be completely desoldered from the PCB.

Ideally, we would want our voltage divider to mimic the voltage of the centered pot. Using our trusty multimeter, we determined that ideal resistance to be about 2.4 k Ω . Not feeling like taking a trip to the electronics store, we rooted around our toolbox to see what we had on hand and determined that a 2.2 k Ω resistor would be close enough, leaving the slack to be taken up by the trim on our radio controller.

To make our voltage divider, we twisted one leg from each resistor together to make a Y, with the common lead going to the center pin on the PCB. The other legs of the Y went to the other pins, and the only tricky part of the procedure was being clean with the soldering and not shorting anything with a solder bridge.

We carefully put the casing back together, making sure that the legs of the resistors did not contact the aluminum casing of the servo. Reassembling the gearbox was straightforward, and once everything was back together it was so indistinguishable from its partial rotation cousins that we had to mark the servo to tell it apart at a glance.

Another option for a full rotation hack that we did not pursue is one that keeps the potentiometer. For some servos, the potentiometer knob can be shortened so that it no longer spins with the servo gears. The shortened knob can be notched so that the potentiometer can still be adjusted without interfering with the rotation of the gearbox. A hole can be drilled in the servo casing so that the pot can still be adjusted. This, however, was not an option with the Savox servo because the pot was situated directly under the main gear. We could have possibly shortened the knob, but replacing the entire pot with the voltage divider avoided any problems of incidental rotation of the knob.

People Livin' in Competition

Now that we had a full rotation Savox servo, we wanted to see how much of its speed and torque we retained in the hack. This could be achieved by lonesome tests of strength and speed, but we are hard pressed to remember an exciting training montage that led up to a climax where the protagonist simply went for a

personal best. We wanted to pit the Savöx head to head against some other servos, so to do that we needed a common platform that could be used to test servos from all walks of life.

To make that platform, we turned to our trusty VEX kit. Given the fairly straightforward nature of hacking the Savöx servo into full rotation, we could have done another one to assist in making a two wheel drive base, but we thought using a single servo would be the best way to test the abilities of the unit.

Using a single servo would head off problems like a lack of coordination between the servos because of trim irregularities, and it would give us the opportunity to use the VEX differential set that we had used back in the September '07 issue. The VEX kit is the perfect platform for rapid prototyping, and we were able to put together a driving base so quickly that we had only heard the word "vener" twice on the Antiques Roadshow by the time we were done.

We used a fixed differential because we thought a drag race and sort of tractor push competition would be easiest to administer. The full rotation servos that the Savöx would compete against would be a servo from the VEX kit itself and a Futaba S3004 that had also been modified for full rotation. The VEX servo was very easy to mate to our drive base, but the small holes on the servo horns for the Futaba and Savöx were not meant to accommodate the square shaft of the VEX kit. We were able to devise a coupler that picked up the center hole on the servo horn and one more hole that needed to be drilled and threaded on the horn. Because the Futaba and Savöx servos have the same splines, we only had to prepare one horn for the coupler. After attaching the necessary electronics, we had a driving base ready for feats of strength and speed.

As with many of our projects, the tiled entryway served as the arena for the tests. For our strength test, we wanted a variable weight for a sort of tractor push. A perfect variable weight was made from a large Tupperware container and several issues of *SERVO Magazine*. We started with a large stack of 'zines in the container and took them away one at a time until

the robot was able to continuously push the container for more than two feet.

The Futaba S3004 was able to push a modest load of two issues of *SERVO*, along with a larger issue of *Nuts & Volts* that had snuck into the stack. The VEX servo was able to shove a larger stack of six issues of *SERVO*, along with the stowaway *Nuts & Volts Magazine*. Finally, the torquey Savöx was able to displace a scale tipping stack of nine issues of *SERVO* and the *Nuts & Volts*.

This result was expected, but the real test would come with our drag race. Torque and speed are normally a trade-

The Death Star was an ARDX kit that went one step too far.




SKU: ARDX Price: \$84.95

The Experimenter's Kit for Arduino features 12 experiments that are step-by-step simple, full-colour template overlays for your breadboard, and every single component you'll need.

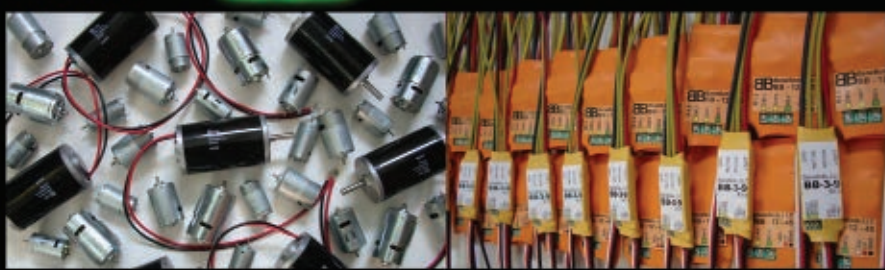
SOLARBOTICS Ltd
www.solarbotics.com 1-866-276-2687

PS- So does that mean that the second Death Star was using a planetary shield shield?




**MOTORS
GEARBOXES
WHEELS
AND MORE**

BB BaneBots BANEBOTS.COM 970-461-8880



ALL ELECTRONICS

C O R P O R A T I O N

THOUSANDS OF ELECTRONIC
PARTS AND SUPPLIES

VISIT OUR ONLINE STORE AT
www.allelectronics.com

WALL TRANSFORMERS, ALARMS,
FUSES, CABLE TIES, RELAYS, OPTO
ELECTRONICS, KNOBS, VIDEO
ACCESSORIES, SIRENS, SOLDER
ACCESSORIES, MOTORS, DIODES,
HEAT SINKS, CAPACITORS, CHOKES,
TOOLS, FASTENERS, TERMINAL
STRIPS, CRIMP CONNECTORS,
L.E.D.S., DISPLAYS, FANS, BREAD-
BOARDS, RESISTORS, SOLAR CELLS,
BUZZERS, BATTERIES, MAGNETS,
CAMERAS, DC-DC CONVERTERS,
HEADPHONES, LAMPS, PANEL
METERS, SWITCHES, SPEAKERS,
PELTIER DEVICES, and much more....

ORDER TOLL FREE
1-800-826-5432

Ask for our FREE 96 page catalog

Firgelli

www.firgelli.com

LINEAR SERVOS



L12-R Linear Servo

Direct replacement for regular rotary servos
Standard 3 wire connectors
Compatible with most R/C receivers
1-2ms PWM control signal, 6v power
1", 2" & 4" strokes
3 - 10 lbs. force ranges
1/4" - 1" per second speed ranges
Options include Limit Switches and
Position Feedback

L12-NXT Linear Servo

Designed for LEGO Mindstorms NXTTM
Plugs directly into your NXT Brick
NXT-G Block available for download
Can be used with Technic and PF
Max Speed: 1/2" per sec.
Pushes up to 5lbs.
2" & 4" strokes



PQ12 Linear Actuator

Miniature Linear Motion Devices
6 or 12 Volts, 3/4" Stroke
Up to 5 lbs force.
Integrated position feedback or
Limit switches at end of stroke
External position control avail.



Available Now At

www.firgelli.com

Twin Tweaks ...

off, so one might expect that the Savöx would be a bit of a slowpoke after seemingly placing all of its eggs in the torque basket.

Such was not the case. The Futaba S3004 plodded across our 86 inch long course in about 6.8 seconds, for a languid pace of about 12.6 inches per second. The VEX servo — a perennial favorite of ours for robotics competitions — zipped along our course in 4.8 seconds for a far more hare-like pace of about 17.9 inches per second. For the grand finale, we outfitted the driving base with the Savöx servo, unsure if it could top the speedy time of the VEX servo. With the explosive kinetic energy of a competitive jumping dog, the Savöx powered bot blazed through the course in 2.9 seconds, for a blistering pace of 29.6 inches per second.

Savöx Talks the Talk and Walks the Walk

Savöx extols the quality of their high performance servo, and we were hopeful that high performance for steering in planes and boats and automobiles could be translated to robotics applications. The full rotation hack was surprisingly smooth despite the initially impenetrable look of the gearbox. With the pin removed and the pot replaced by a fixed value voltage divider, the SC-1258TG makes an excellent full rotation servo that demonstrated the speed and torque ideal for competitive robotics applications. We think it quite likely indeed that the Savöx servos will be making future appearances in our projects; projects supported by the classic fundamental full rotation servo hack. **SV**

Robotics Showcase

Recycling & Remarketing High Technology
WEIRDSTUFF[®]
WAREHOUSE

Software, Computers, Electronics, Equipment, Doo-hickies

384 W. Caribbean Dr.
Sunnyvale, CA 94089

Mon-Sat: 9:30-6:00 Sun: 11:00-5:00

(408)743-5650 Store x324



**WE BUY AND SELL EXCESS
& OBSOLETE INVENTORIES!**

FREE COMPUTER RECYCLING

We recycle computers, monitors, and
electronic equipment. M-Sat 9:30-4:00



GREAT DEALS!

Hi-tech items, electronics
test equipment, and more!

GIANT AS-IS SECTION

10,000 sq. ft. of computers, electronics,
software, doo-hickies, cables, and more!



also check out our...

eBay Store
stores.ebay.com/WeirdStuff-Inc

WWW.WEIRDSTUFF.COM

Esduino12

- 9S12C 16-bit microcontroller in Arduino form-factor!
- 32K or 128K Flash
- optional USB interface
- low-cost Xbee plug-in option

Use with any
Arduino shield!



From
\$39

Easy object-based
Programming with nqBASIC!
Advanced programming in C
with CodeWarrior



Four new proto shields!

TechnologicalArts.com

**RECOMMENDED
websites**

www.Savoxusa.com

www.acroname.com/robotics/info/ideas/continuous/continuous.html

The *SERVO* Webstore

Attention Subscribers ask about your discount on prices marked with an *

CD-ROM SPECIALS



7 CD-ROMs & Hat Special
Only \$ 149.95 or \$24.95 each.
www.servomagazine.com



On Sale Now!
\$24.95
we'll ship for FREE!

ROBOTICS

PIC Robotics by John Iovine

Here's everything the robotics hobbyist needs to harness the power of the PICMicro MCU!

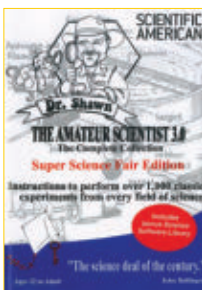
In this heavily-illustrated resource, author John Iovine provides plans and complete parts lists for 11 easy-to-build robots each with a PICMicro "brain." The expertly written coverage of the PIC Basic Computer makes programming a snap – and lots of fun.

\$24.95

The Amateur Scientist 4.0 The Complete Collection by Bright Science, LLC

There are 1,000 projects on this **CD**, not to mention the additional technical info and bonus features. It doesn't matter if you're a complete novice looking to do your first science fair project or a super tech-head gadget freak; there are enough projects on the single CD-ROM to keep you and 50 of your friends busy for a lifetime!

Reg \$26.95 Sale Price \$23.95



Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists by Dustyn Roberts

In *Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists*, you'll learn how to successfully build moving mechanisms through non-technical explanations, examples, and do-it-yourself projects – from kinetic art installations to creative toys to energy-harvesting devices. Photographs, illustrations, screenshots, and images of 3D models are included for each project.

\$29.95*

Build Your Own Humanoid Robots

by Karl Williams
GREAT 'DROIDS, INDEED!

This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than \$300. Together, they form the essential ingredients for making your own humanoid robot.

\$24.95*



Robot Programmer's Bonanza by John Blankenship, Samuel Mishal

The first hands-on programming guide for today's robot hobbyist!

Get ready to reach into your programming toolbox and control a robot like never before! *Robot Programmer's Bonanza* is the one-stop guide for everyone from robot novices to advanced hobbyists who are ready to go beyond just building robots and start programming them to perform useful tasks.

\$29.95



Robotics Demystified

by Edwin Wise

YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS!

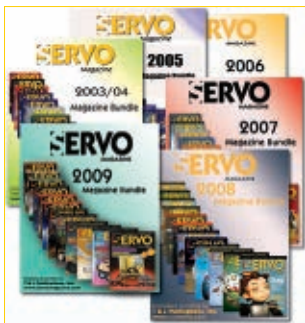
Now anyone with an interest in robotics can gain a deeper understanding – without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots! **\$19.95**



**We accept VISA, MC, AMEX,
and DISCOVER
Prices do not include shipping and
may be subject to change.**

To order call 1-800-783-4624

SERVO Magazine Bundles



Published by T & L Publications, Inc.

\$57
per bundle

Save \$10
off the
normal
price!!

Now you can get one year's worth of all your favorite articles from *SERVO Magazine* in a convenient bundle of print copies. Available for years 04, 05, 06, 07, 08, and 09.

LEGO MINDSTORMS NXT Idea Book

by the Contributors to
The NXT Step Blog

If you're serious about having fun with LEGO® robotics, you've come to the right place. The team behind The NXT STEP blog — the authoritative online source for MINDSTORMS® NXT information and advice — has packaged its considerable skills and experience in this book. Inside, you'll find some of the team's best ideas for creating cool and sophisticated models, including instructions for eight robots you can build yourself.

Reg \$29.95 Sale Price \$24.95

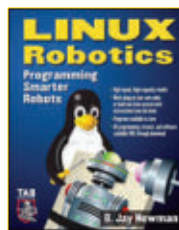


Linux Robotics

by D. Jay Newman

If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need.

\$34.95



CNC Machining Handbook: Building, Programming, and Implementation

by Alan Overby

NEW!

The *CNC Machining Handbook* describes the steps involved in building a CNC machine and successfully implementing it in a real world application. Helpful photos and illustrations are featured throughout. Whether you're a student, hobbyist, or business owner looking to move from a manual manufacturing process to the accuracy and repeatability of what CNC has to offer, you'll benefit from the in-depth information in this comprehensive resource.

\$34.95



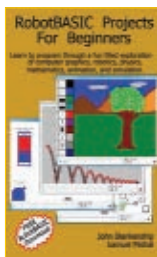
SPECIAL OFFERS

RobotBASIC Projects For Beginners

by John Blankenship, Samuel Mishal

If you want to learn how to program, this is the book for you. Most texts on programming offer dry, boring examples that are difficult to follow. In this book, a wide variety of interesting and relevant subjects are explored using a problem-solving methodology that develops logical thinking skills while making learning fun. RobotBASIC is an easy-to-use computer language available for any Windows-based PC and is used throughout the text.

Reg. Price \$14.95 Sale Price \$9.95



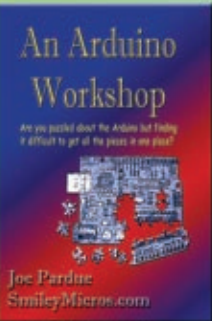
Technology Education Package for Everyone Starting in Electronics

This lab — from the good people at GSS Tech Ed — will show you 40 of the most simple and interesting experiments and lessons you have ever seen on a solderless circuit board. As you do each experiment, you learn how basic components work in a circuit. Along with the purchase of the lab, you will receive a special password to access the fantastic online interactive software to help you fully understand all the electronic principles. For a complete product description and sample software, please visit our webstore.

Regular Price \$79.95

Subscriber's Price \$75.95

SPECIAL OFFERS



An Arduino Workshop
Are you puzzled about the Arduino but finding it difficult to get all the pieces in one place?
Joe Pardue
SmileyMicros.com
Book \$44.95

Puzzled by the Arduino?

Based on the *Nuts & Volts* Smileys Workshop, this set gives you all the pieces you need!

Book and Kit Combo \$124.95



Kit \$84.95

For more info on this and other great combos, please visit: <http://store.nutsvolts.com>



Getting Started with PIC's
A Collection of 2008 Nuts & Volts Magazine Articles
Chuck Hellebricks

Enter the world of PICs & Programming with this great combo!

Combo Price \$175.95



Embedded C Programming
Volume 2
Stephen's Books

For complete details visit our website @ www.nutsvolts.com

Mechanisms and Mechanical Devices Sourcebook

by Neil Sclater, Nicholas Chironis
Over 2,000 drawings make this sourcebook a gold mine of information for learning and innovating in mechanical design. Overviews of robotics, rapid prototyping, MEMS, and nanotechnology will get you up to speed on these cutting-edge technologies. Easy-to-read tutorial chapters on the basics of mechanisms and motion control will introduce those subjects to you. **Reg \$89.95 Sale Price \$69.95**



Forbidden LEGO

by Ulrik Pilegaard / Mike Dooley
Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse. **Reg \$24.95 Sale Price \$19.95**



PROJECTS

The SERVO Buddy Kit



An inexpensive circuit you can build to control a servo without a microcontroller.



For more information, please check out the May 2008 issue or go to the **SERVO** webstore.

Includes an article reprint.

Subscriber's Price **\$39.55**

Non-Subscriber's Price **\$43.95**

16-Bit Micro Experimenter Board



Ready to move on from eight-bit to 16-bit microcontrollers? Well, you're in luck!

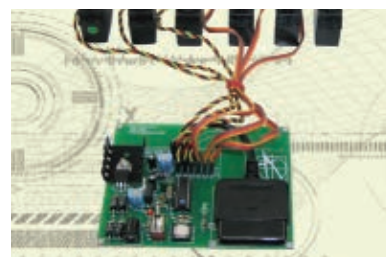
In the December 2009 *Nuts & Volts* issue, you're introduced to the 16-Bit Micro Experimenter.

The kit comes with a CD-ROM that contains details on assembly, operation, as well as an assortment of ready-made applications. New applications will be added in upcoming months.

Subscriber's Price **\$55.95**

Non-Subscriber's Price **\$59.95**

PS2 Servomotor Controller Kit



This kit accompanied with your own PlayStation controller will allow you to control up to six servomotors.

Includes all components and instruction manual.



For more information, please see the February 2011 edition of *SERVO Magazine*. Assembled units available!

Subscriber's Price

\$79.95

Non-Subscriber's Price

\$84.95



The NXT Big Thing #12



By Greg Intermaggio



Howdy doody, Robotsons!
We're celebrating one year
of awesome LEGO
MINDSTORMS NXT projects
with this column!

This month, we'll take our
last project to the NXT level.

Using the NXT Bluetooth
capability and two gyroscopic
sensors from HiTechnic, we'll
learn to control Eddie
by flailing our arms as if we
were playing Wii!



Because this is a bigger programming challenge than we've ever seen, I couldn't fit it all in one article! This will be the first of a two-part series. Be sure and pick up next month's edition of *SERVO Magazine* for the conclusion.

This article will assume that you:

- Have already built Eddie 2.0, or a similar driving base that uses tank steering. Eddie 2.0 building instructions can be found in the January '11 issue.
- Have already purchased two HiTechnic gyro sensors, (available at HiTechnic.com) and imported the gyro sensor block into the NXT software. This month, we'll only be using one, but next month we'll use one for forward/backward and another for left/right.
- Have a second NXT to use as a remote control.

Onward!

Introducing the Challenge

This will be our coolest and most difficult challenge yet! The idea is that we want to detect angular rotation from the HiTechnics gyro sensors, send that data over Bluetooth to the other NXT, and then use it to control Eddie 2.0 (or your own driving robot).

If that wasn't complicated enough, we're also going to have to use our own programming hack to turn the sensor readings into something we can use, but more on that later.

Exploring the Gyro Sensors

Let's get started by understanding the sensors we'll be using. HiTechnics gyro sensors measure angular movement

in degrees per second. This means that if the sensor is not moving, it will read 0. If the sensor is spinning at 1 RPM (rotation per minute), that means that it's spinning 360 degrees every 60 seconds; $360/60 = 6$ so if the gyro sensor is moving 1 RPM, it will read six degrees per second.

Let's write a quick program to better visualize how the sensor readings work.

Go ahead and test the program. You'll want to plug your gyro sensor into port 3, or whatever port you specified. Run the program, and swing the sensor side to side. You'll see that the numbers on your NXT display will increase in absolute value the faster you move the sensor, then decrease in absolute value the slower you move.

Gyro Test Program 1 Instructions

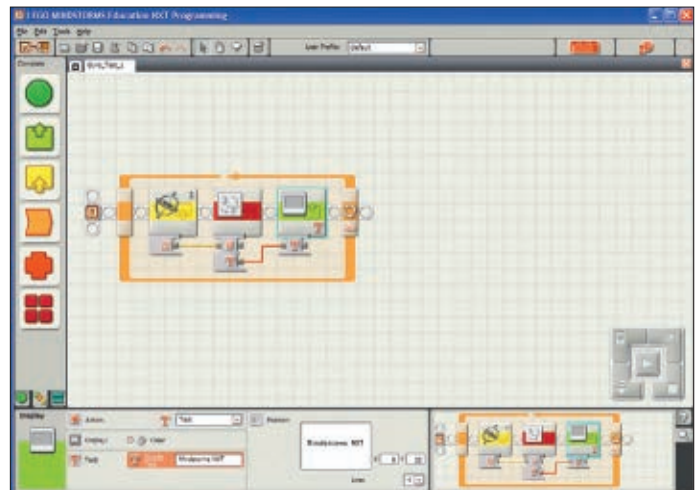


Figure 2. From the advanced palette, add a number-to-text block. Then add a display block from the common palette. Set the display action to Text. Finally, wire the Raw Value output of the gyro sensor to the input of the number-to-text block, and wire the output of the number-to-text block to the text input of the display block.

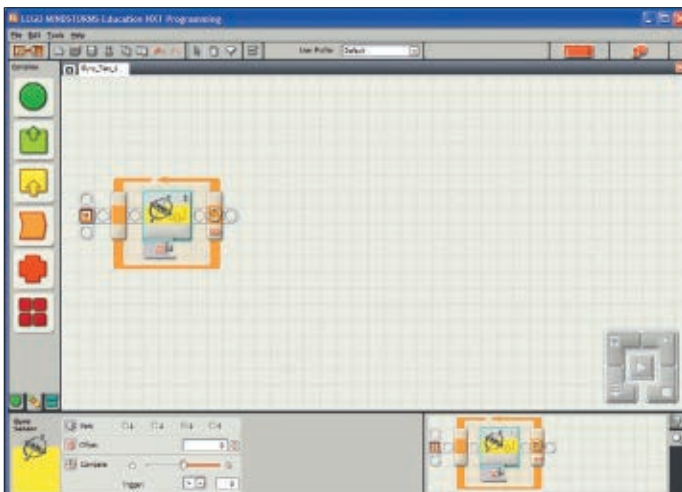


Figure 1. Create a new program called Gyro_Test_1. Add a loop, and inside the loop, add a gyro sensor block.

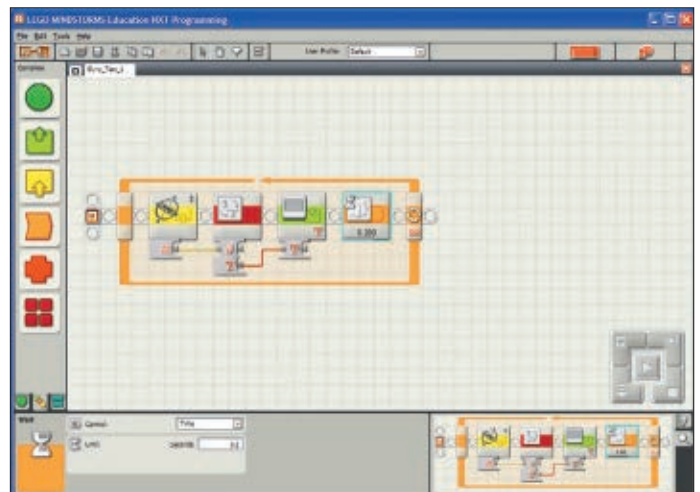
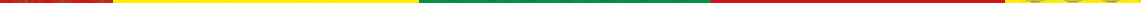


Figure 3. Add a Wait For block and set it to .2 seconds (that's 1/5th of a second – not two seconds!).

The way we'll do it is to take an average reading from our gyro sensor when it isn't moving, then use that average to offset the output. So, if our average reading is 7 when the sensor isn't moving, we'll subtract 7 from each reading to make the sensor read 0 instead. Here's how:

The screenshot displays the LEGO MINDSTORMS Educative NXT software interface. The main workspace contains a sequence of blocks: a loop block, a motor block, a wait block, and a sensor block. A dialog box titled "Edit Variables" is open, showing a table with columns "Name" and "Type". The table contains three rows: "loop" (Type: loop), "wait" (Type: wait), and "sensor" (Type: sensor). The "sensor" row is selected. Below the table are buttons for "Create", "Delete", and "Reset". The bottom status bar shows the current step in the sequence.

The screenshot shows the LEGO Mindstorms NXT software interface. The main workspace contains a sequence of blocks: a 'Motor' block (yellow), a 'Wait' block (grey), a 'Light' block (green), a 'Wait' block (grey), a 'Motor' block (yellow), a 'Wait' block (grey), a 'Light' block (green), a 'Wait' block (grey), and a 'Motor' block (yellow). The blocks are connected in a linear sequence. The interface includes a toolbar on the left with various icons for block types, a top menu bar with 'File', 'Edit', 'View', 'Tools', 'Help', and 'About', and a bottom status bar with 'Status' and 'Current'.

[illegible]

Go ahead and run your program. You should notice that the values output from the gyro sensor are much closer to 0 when it's not moving. Okay, let's get on to the real challenge ...

Hacking the Gyro

To make our gyro remote control, we first need to understand the difference between the sensor output and the values we need. Like we learned previously, the gyro sensor gives its reading in degrees per second. That means that while it's moving, it will either give a positive or a negative reading. When it's not moving, the sensor reads 0.

This is a big problem for us because we want to be able to swing our arm forward once to make Eddie move continuously forward. If he stops moving when our arm stops moving, we'll get real tired real quick!

So, how do we get around it? It's not as hard as you might think. The important thing to remember is that if the sensor moves clockwise, it will give a positive reading, and if it moves counter-clockwise, it will give a negative reading. This means that if it's going fast clockwise for one reading at, say 360 degrees per second, then reverses its movement to go -360 per second for another reading, when added together, the two cancel each other out and make zero.

Let's think this through a bit more. Say I swing my arm forward while holding the gyro sensor. We'll pretend it only takes one reading while my arm is swinging (but really it takes readings up to 300 times every second!). Let's also say the one reading it takes is 90 degrees per second. If we add that reading to a variable, that variable now reads 90.

Our variable is now set to 90, and I've stopped my arm from moving, so our sensor reads 0. If we add 0 to 90, our variable stays the same. If we were sending that information to Eddie, he could keep moving.

But what if we want him to slow down, or even back up? Simple! If I already swung my arm forward and got the reading of 90, all I need to do is swing my arm backwards the same distance to get -90. If we add -90

Gyro Test Program 3 Instructions

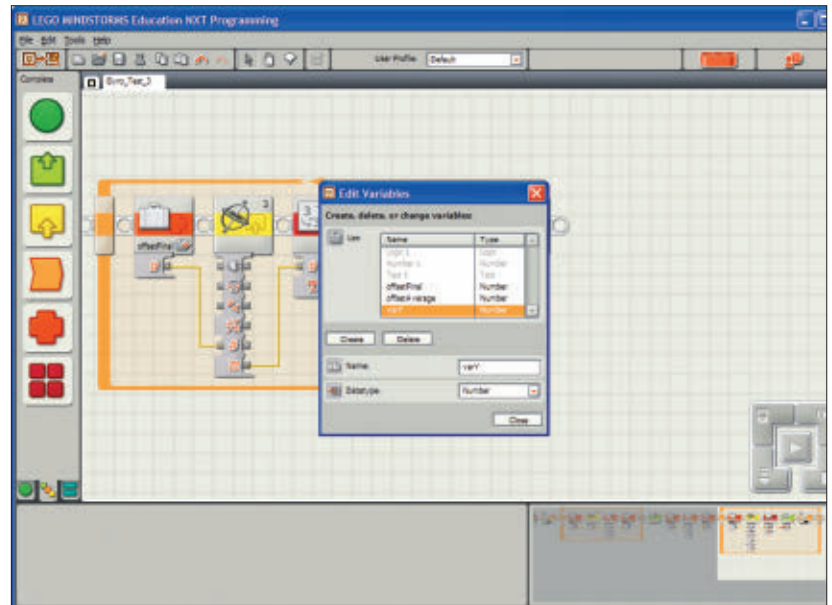


Figure 1. Start by saving your old gyro test program as Gyro_Test_3, so you'll have a backup just in case! Next, add a number variable called "varY" — we'll call it that because it will eventually take movement readings in the Y axis, and control Eddie's forward and backward movement.

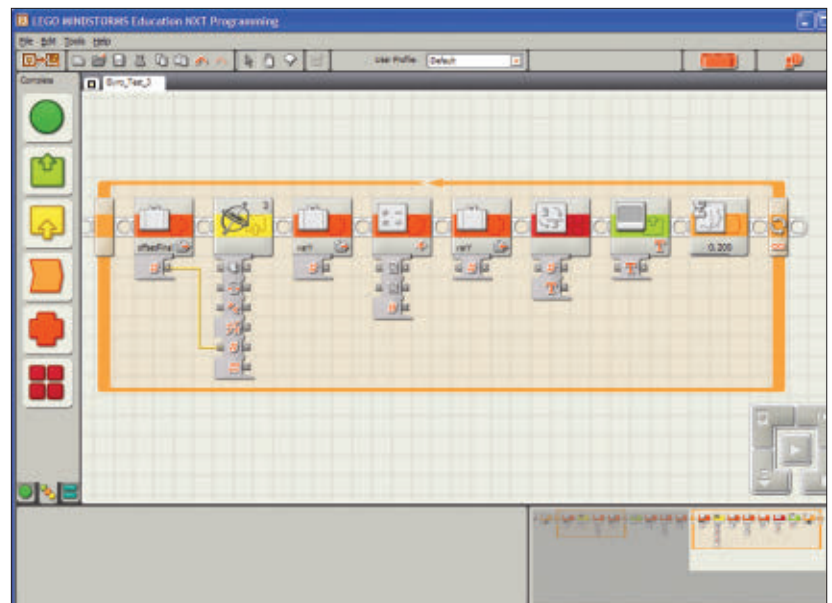


Figure 2. Okay. Now we're going to add a Read Variable block just after reading the gyro sensor and set it to varY. After that, we add a Math block and set it to addition, and a Write Variable block also set to varY. Note that these all go between the gyro sensor reading and the Number to Text block. You'll probably want to delete your old wiring like I did to make it easier to work with.

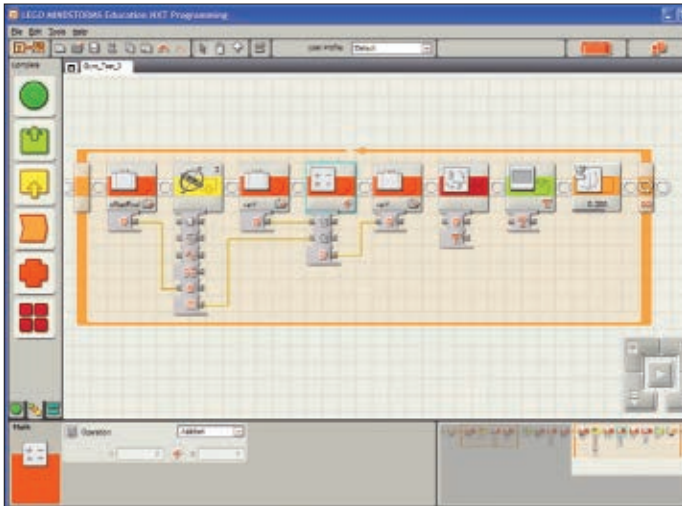


Figure 3. Wire the Raw Value from the gyro sensor and the Read varY block to the Addition block, then wire the result to write to varY. This means we'll be adding together the current varY and the current gyro sensor reading, and saving that result to varY.

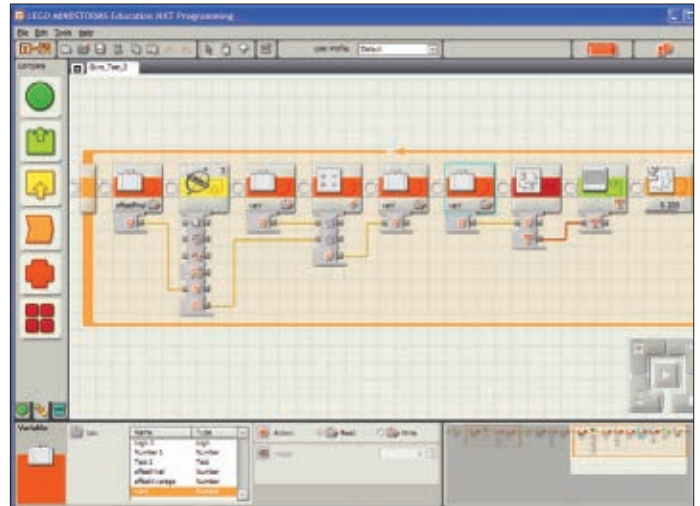


Figure 4. Finally, add a Read varY block and wire it to the Number to Text block, with the result wired to the Display block as before.

to 90, we get 0, and Eddie stops moving. If I swing my arm back further still, we get a negative number that we can use to tell Eddie to go backwards.

Okay. Let's write the program (see **Gyro Test**

Program 3 Instructions).

Go ahead and test this program. If all goes well, you should see that turning the sensor clockwise will add to the varY variable, and turning the sensor counter-clockwise will subtract. This data will be perfect for us to make Eddie roll based on our arm movements.

Wrapping Up

That was a heck of a programming challenge! Not only did we learn a new sensor, we created a program that would re-interpret the data from the gyro sensor and turn it into something we can use.

Next month, we'll add in a second gyro sensor for the x-axis, Bluetooth communication with Eddie, and the program that will make Eddie move based on our gyro sensor readings. See you next month! **SV**

HiTechnic

The only complete range
of LEGO Certified sensors and
accessories for LEGO MINDSTORMS

NXT Color Sensor V2

NXT Angle Sensor

For a complete list
of all our products
please visit our
website.

Email us: sales@hitechnic.com

www.HiTechnic.com

About the Author

Greg "LEGO" Intermaggio lives in the Bay Area, CA, where he runs a business called Techsplosion, bringing hands-on science to all ages and walks of life. In his spare time, Greg

likes to unicycle, juggle, unicycle juggle, and battle killer robots!

More information about Greg can be found at **Intermaggio.com**.

More information about Techsplosion can be found at **Techsplosion.org**





Then and NOW

ROBOTS: FORM OR FUNCTION?

by Tom Carroll

There were some interesting comments on the various threads of the Seattle Robotics Society's server in early April about what was more important: how a robot looked or how well it performed. Interestingly, both of these attributes are subject to personal preferences, especially the aesthetics of a particular robot. Most of us are quite content to build a robot that looks as good as we can make it, and have all of the physical and programmed functions work how we want them to work. At some point — as we scavenge or craft all of the parts of the robot that we can — we settle on the robot's final appearance. The robot's physical look is quite often either the result of a lot of time-consuming machining, carving, and polishing, or it is the outcome of what nice looking 'things' we can find to put on our creation. We can add circuits, sensors, motors, wheels, arms, claws, batteries, and even external lights that we buy or find somewhere within our stock of saved parts.

As far as function goes, we can test the many lines of code that we spent hours trying out on the bench, debugging, deleting, adding new code, and finally, loading it into our microcontroller. We can sit down with a CAD program and lay out the chassis and add the many motors, sensors, appendages, drive wheels, and external shell to approximate our final design. Most of the time, we have to measure the sizes of larger components such as motors, batteries, and drive systems that we have already located or purchased, and figure out where we need to place and mount them to have them function as we planned. Without going into complex differential equations, we can use finite element analysis programs to hone kinematics and the robot's operations over different terrains to determine motor and gearbox sizes and configurations in the design process. Of course, robot design is certainly more complex than the few steps I just mentioned.

Robot Beauty and Function

It has been said that "beauty is in the eye of the beholder." Some people may look

at a particular robot and feel that it is truly a work of art. They can view sleek lines, hidden motors, wires, and circuitry, and see a magnificent robot. Many builders have tried and succeeded to build a copy of the Johnny Five robot from the film, "Short Circuit," shown in **Figure 1**. In that movie, there were no wires snaking up the arms of the robot and all of the articulated joints hid the internal components.

FIGURE 1. Johnny Five from the film 'Short Circuit.'



Many people have told me that they feel the Johnny Five robot was one of the most beautiful of all movie robots for that reason. Yes, it was a studio action prop, remotely-controlled by multiple special-effects operators on the sidelines, but the electronics and mechanics made Johnny Five one of the most complex action props ever used. These days, studios find it cheaper to construct computer-generated robots, such as the NS-5s from the Will Smith film, *i-Robot* — especially when the script called for a thousand robots as in **Figure 2**.

In a way, we can view function in the same light. People who have no idea just how hard it is to build a complex robot may laugh if they happen to see a bipedal humanoid take a few wobbly steps and then

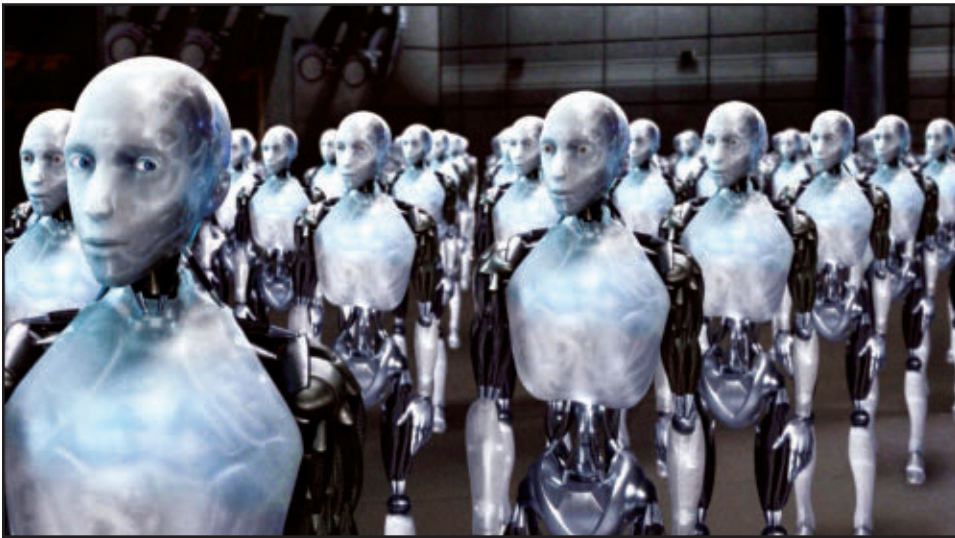


FIGURE 2. NS-5 robots from the film, "i-Robot."

fall flat on its face. They don't realize the complex programming involved, with inputs from accelerometers, gyros, and other sensors. We view the process of walking for humans the same way, not knowing about the ear's semi-circular canals that act as dual-axis accelerometers and gyros.

Figure 3 shows another very functional robot — DLR's 'Rollin' Justin' from the German Aerospace Center. Justin was built as a prototype astronaut's aide for space applications, much the same as NASA's Robonaut II. This latest version has a wheeled base for terrestrial travels. Justin recently made the route of YouTube videos as he was seen catching one and two balls at the same time, with a catch accuracy of 89%. Afterwards, he was seen pouring himself a cup of coffee. Some people may say "big deal" to that, but I'd be willing to bet that a person making those

kinds of comments could not build any of the robots mentioned in this article.

The Design Process

Most of us do not use FEA or CAD programs in our machine's design process. Instead, we might start with a flat piece of metal "that looks just about the right size" and we begin the design process by laying the different parts that we've collected on this base. For a differential-drive robot, we might place two gear motors (or servos for a small robot) centered on the sides of the base and determine how we'll mount these motors and attach the wheels. We then might add two spring-loaded castors fore and aft.

At this point, we have a basic robot base; the addition of the rest of the components is a bit easier. The heavy battery is usually placed down low and centered for stability. The processor, H-bridges, sensors, and any appendage can be mounted on the main base or on a smaller base on top of four long spacers. As for arm designs, we can look into our pile of parts and find a large servo or gearmotor of a certain torque and speed, and 'backward engineer' the design process to fabricate an arm that will lift a certain weight. If we need more lifting capacity, we either shorten the arm's length or gear down the speed to obtain more torque. (Again, the actual design process is a bit more complex than this.)

Jim Hill's Charlie

I'd like to go back to the early '80s and discuss a robot made by a friend of



FIGURE 3. DLR's Rollin' Justin space robot.

FIGURE 4. Jim Hill's Charlie in *Popular Mechanics*.



FIGURE 5. Jim Hill and Charlie. (Photo by Roger Ressmeyer.)



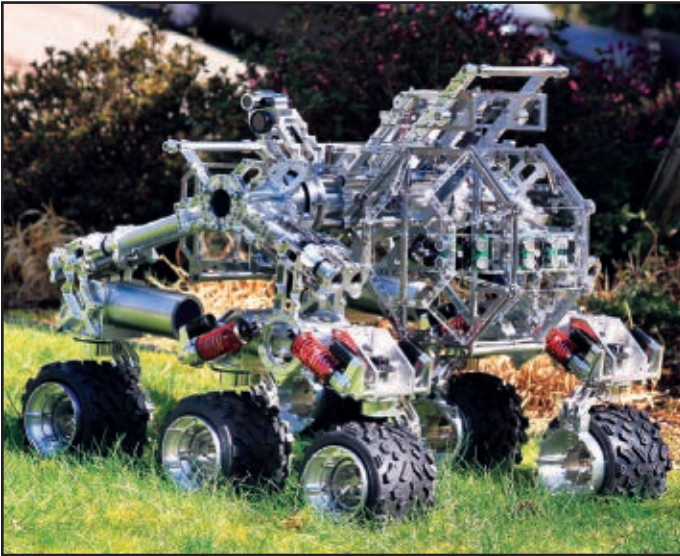


FIGURE 6. Ken Maxon's robot, Sojourner.

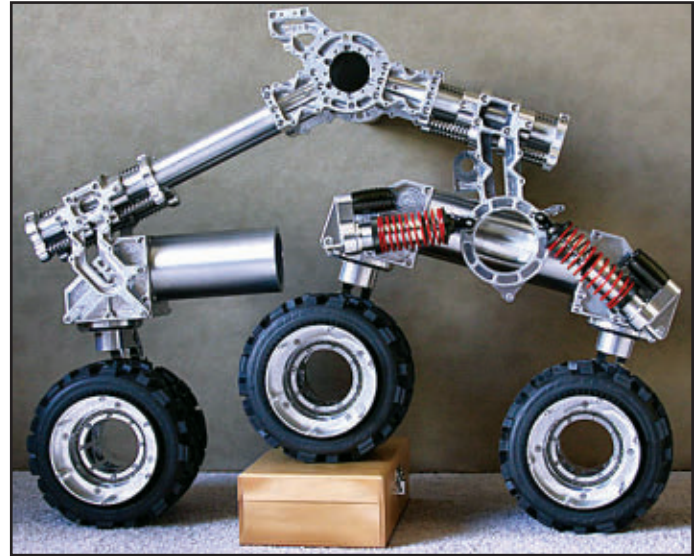


FIGURE 7. Rocker bogey suspension.

mine, Jim Hill. Jim was a parts manager for a Chrysler dealership in West Covina, CA and built his robot entirely by using hand tools; the only power tool he used was a 120 VAC drill. I had long admired Jim's creation and used it as the cover photograph (shown in **Figure 4**) for a *Popular Mechanics* story I wrote on home-built robots back in 1984. Jim concentrated on both form and function in creating Charlie. All the metal was cut by a hacksaw and then filed smooth. The complex movements of the arms and claws were accomplished by 12 VDC electric car seat motors within the robot's body, and the rotary motion delivered to the different linear actuators was done by flexible shafts. The arm's joints used surplus aircraft flap actuators that acted like human muscles.

Jim is shown in **Figure 5** beside Charlie in a photo by Roger Ressmeyer. Jim's robot was not a simple, servo-driven tabletop robot and was so heavy that he needed a trailer to bring it down to my house in Long Beach, CA, for the photographs. I have frequently used his robot as an example of how virtually anybody with a

mechanical bent and basic hand tools can build a great robot. Charlie is an early example of a fine robot exhibiting both great aesthetical form and function in one machine.

Ken Maxon's Robots

Jumping to the present, the robot that started the series of comments on the Seattle Robotics Society's list I mentioned earlier was Ken Maxon's Sojourner shown in **Figure 6**. It is quite obvious that this robot is probably second to none in overall aesthetics (at least from a machinist's point of view), and that includes any factory-built machine. Ken named it Sojourner; not necessarily to copy the name of the first NASA Mars-roving robot, but an appropriate name for a great robot that uses the NASA JPL-designed six-wheel rocker-bogey system. This suspension is shown being tried out in **Figure 7**. The few finished parts lying on the table in **Figure 8** illustrates just how much machining was required in the

FIGURE 8. Parts being machined.

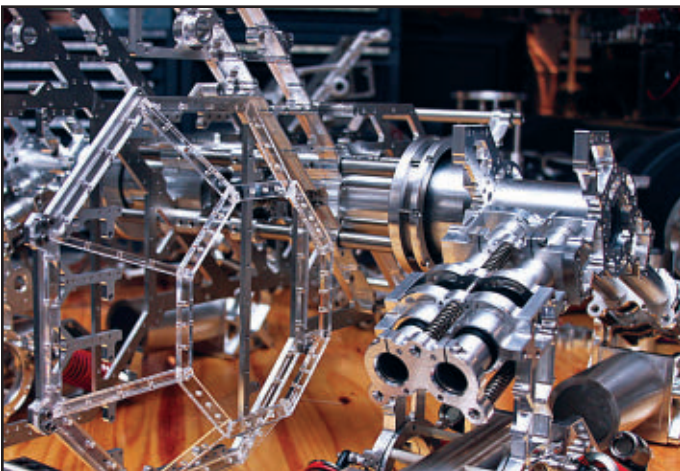


FIGURE 9. Ken's electronics workshop.



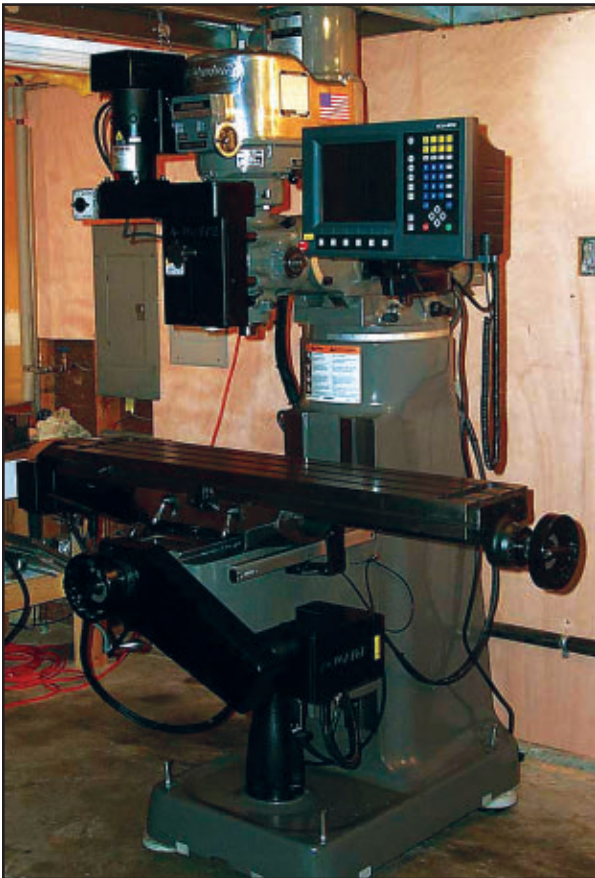


FIGURE 10. Ken's new milling CNC machine.

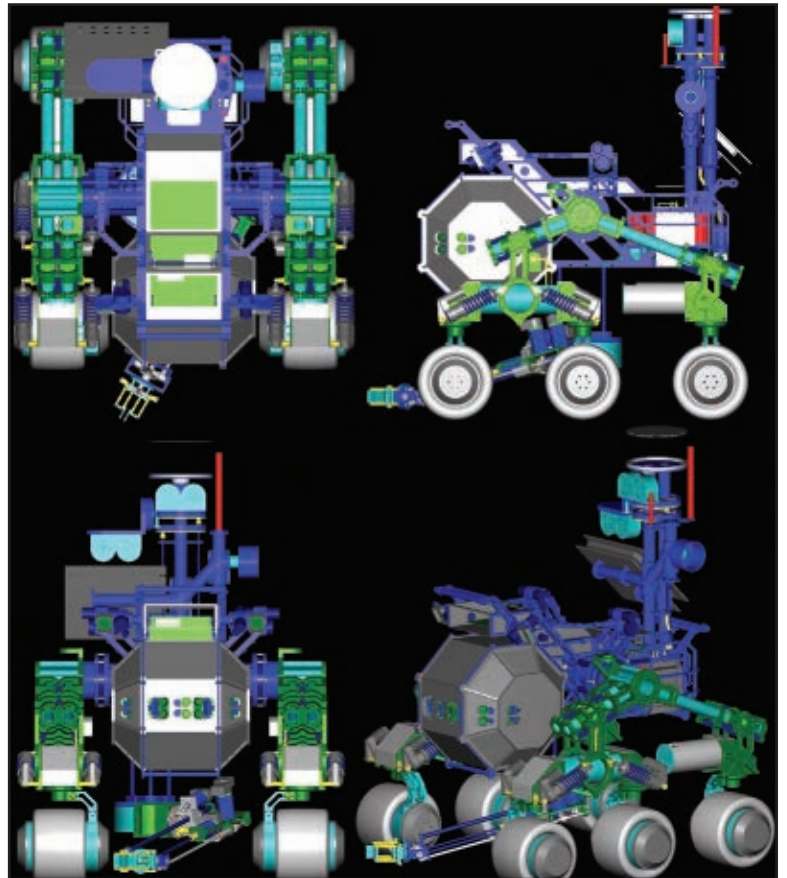


FIGURE 11. CAD models of Sojourner.

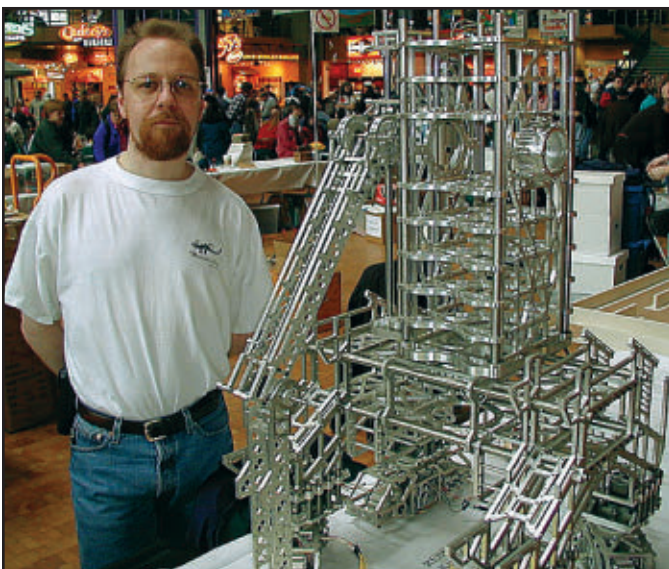
construction process.

Ken's basic goal was to design a robot to autonomously navigate the 3/4 mile track around the block where his home is located. A robot of this complexity is no easy weekend build. He started the project in March 2006 and mid-way took a one year sabbatical from the task. Unlike the flatter Mars rover that was covered with solar

cells for power, Ken's robot runs on batteries and stands higher, has ultrasonic distance sensors that operate in air; and is possibly the most intricate and well-designed personal robot that I've ever seen — and I've seen thousands.

Yes, Ken's creation is an extreme example of form, but he has not forgotten the need to have superb function in the robot. Photos on his site illustrate many of the custom circuit boards that he used.

FIGURE 12. Ken Maxon with 'Project Max' at Robothon.



A section of his electronics fabrication shop is shown in **Figure 9**. After using an imported 'mill-drill' milling machine for seven years, Ken purchased a professional 3,800 lb CNC mill shown in **Figure 10**. He used CAD software to form the basic

FIGURE 13. Dave Shinzel's Loki looking up at the camera.

models shown in **Figure 11**.

As with almost every home-built robot, Sojourner is a work-in-progress. Ken had previously exhibited another one of his creations at the SRS Robothon several years ago which is shown in **Figure 12**. You can follow the steps he used in designing the Sojourner robot, see the CAD models, follow the various steps in machining the complex gears, housings, and structural components, and see the parts as he fits them together. I personally spent hours at his site. Go to Max's Little Robot Shop at www.users.qwest.net/~kmaxon.

Dave Shinsel's Loki

Several years ago, I met Dave Shinsel at a Portland Robotics Society meeting where he was demonstrating his most unique robot, Loki (as seen in **Figure 13**). One thing that immediately struck me when I first saw the robot was a very capturing personality using Loki's swiveling head. This endearing form was accomplished by well-designed functions. Loki seemed to me to have that same sort of personal individuality as Johnny Five did. As Dave says, "Loki" is named after the Norse god of Mischief since with robotics "anything that can go wrong, will go wrong ... and usually break something important." Loki seemed like an appropriate name."

I have mentioned Loki before on several occasions, discussing robot evolution and capabilities. Shinsel has since added a second arm that is shown in **Figure 14**. It was developed so the robot can manipulate things in his environment. The configuration is entirely different from the left hand, and makes the robot that much more useful when it encounters a unique object that requires manipulation. That is something few anthropomorphic robot builders ever design into their machines. Dave states: "The new "hand" is built specifically for opening door knobs and is a work-in-progress."

Dave used a lot of components that many other robot experimenters have used in their designs, such as a laptop computer as the main processor, peripheral PIC processors, and computer web cams for eyes. The robot can use these simple cameras to track, identify, and guide one of the manipulators to grasp an object. It is a bit hard to see the laptop's configuration in the photograph but the LCD display is in constant view at the top of the chest, and the



FIGURE 14. LOKI's new hand.



FIGURE 15. Willow Garage's PR-2.

keyboard section is covered except during programming. Dave used two PIR sensors for motion detection, two ultrasonic ranger modules, four bumper switches, an electronic compass, and motion encoders on the arms and wheels. The arms and head motions are driven with Robotis Dynamixel AX-12 rotary actuators.

Though Loki is totally autonomous, Dave can also control the robot via a standard Wii hand controller. However, it was not the robot's stunning ability to interact with people, objects, and its surroundings that impressed me the most; it was Dave's use of common materials to make his home-made robot look like a factory-built product. I've used Loki as an example of a robot's form, but Dave's robot is possibly a better example of an amazingly functional robot, as well. Dave can be reached at shinseljunk@verizon.net, or Google "Loki" to view some great videos.

The Willow Garage PR-2

The PR-2 robot shown in **Figure 15** from the Silicon Valley company, Willow Garage, just may be the most extreme robot designed with both the ultimate in form and function. When I first heard of the PR-2 and some of the other Willow Garage projects, I wondered just what WG was up to. WG was formed by ex-Google employee, Scott Hassan in Menlo Park, CA, back in 2006, and he tapped Steve Cousins as the president and CEO.

"Willow Garage is a robotics research lab and technology incubator devoted to developing hardware and open source software for personal robotics applications,"

Steve told me last year when I visited their facility. The PR-2 certainly proved to me to be more than a beautiful robot as it can do far more than just open doors and plug itself into a socket for 'dinner.'

At around \$280K to \$400K apiece (depending on who you ask), these robots are not for everybody, nor are they to be built with the average budget. WG has actually sold a few since they 'gave' 11 to various universities around the US and the world. Sixteen research labs are finding PR-2 the ideal research platform.

The Personal Robot 2 was not designed as a product to sell to everyone but as a research platform to be used by serious robotics engineers to develop viable platforms that may eventually be used in homes as a personal assistant robot. With a 1.3 kW Li-Ion battery driving the robot's mechanical systems and a pair of Intel Xeon quad-core i7 processors operating on the ROS open source language, PR-2 is a powerful machine to be reckoned with. The unique 8-DOF arms each have a dual parallel claw end-effector, and the arms are spring-balanced and back-drivable to allow them to remain in position — even with no power applied. What I found so appealing when I first saw the robots in person were their extraordinarily-crafted exterior and the unique arm design.

Willow Garage Delves into New Areas of Robotics

WG is not resting its laurels just on the success of the flashy and functional PR-2. They have branched out into other areas of robotics. Their prototype Texai robots may not have the polished form of PR-2, but they have proven to

be quite functional in providing telepresence capabilities for companies that are widely spread apart.

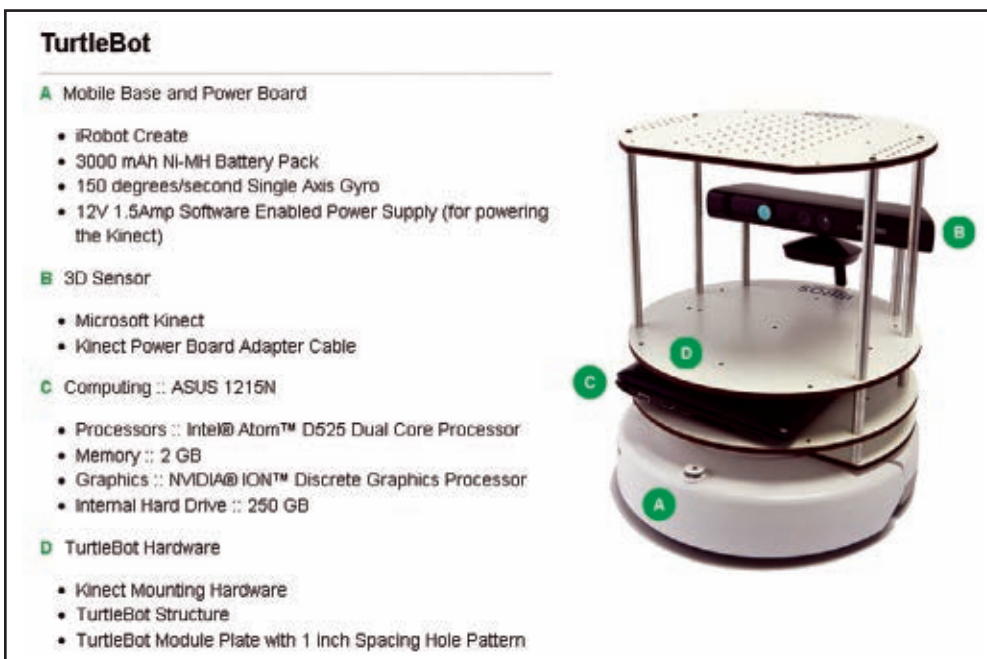
One of the newest products that WG plans on selling to the robotics community (may be as soon as this summer) is what they call the TurtleBot (shown in **Figure 16**). This experimental robot makes use of two very popular technologies that robot developers have been using for a while: the iRobot Create robot base and the Microsoft Kinect vision system for the X-Box 360.

The TurtleBot core kit costs \$500 and includes:

- USB Communications Cable
- TurtleBot Power and Sensor Board
- TurtleBot Hardware
- Microsoft Kinect Sensor
- TurtleBot to Kinect Power Cable
- USB Stick TurtleBot Installer

Adding the iRobot Create, a camera system, an additional battery, and a separate computer brings the total up to \$1,200. In a company blog post, Willow Garage said that since its "mission is to bring personal robotics to the home, we feel that the time is ripe to put a low cost, personal robot kit" in the hands of hobbyists and developers. The complete system can act as a mini security robot or a test platform for developers, and it's capable of taking photos of its environment and downloading them to the web. The kit requires no complex soldering and the open source software is available free on the Internet. WG was smart to use the existing and powerful Kinect sensor, as well as the proven Create robot platform. I feel that the TurtleBot may not have the sleek lines and form like the PR-2, but it is WG's first effort to bring their world-class technology function to robot hobbyists and developers.

FIGURE 16. Willow Garage's Turtlebot.



Final Thoughts

I've covered just a few robots that I feel either exhibit exemplary form or function, or both. I would like for people to look at any robot they might see and look for the best characteristics in each area, even if they may not meet people's particular standards.

If you like how your robotic creation ended up appearing and it performs just the way you designed it to, you should be proud of what you designed and built. **SV**

Tom Carroll can be reached at TWCarroll@aol.com.

ROBO-LINKS

THE ORIGINAL SINCE 1996
PCB-POOL
 Beta LAYOUT

- Low Cost PCB prototypes
- Free laser SMT stencil with all Proto orders

WWW.PCB-POOL.COM

AndyMark
 Inspiring Mobility



www.andymark.com

Ultrasonic Ranging is EZ

- High acoustic power, auto calibration, & auto noise handling makes your job easy.
- Robust, compact, industrial (IP67) versions are available.

www.maxbotix.com

RobotShop
 .com



Pololu
 Robotics & Electronics
WWW.POLOLU.COM




INVEST in your BOT!



HiTEC

12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecrod.com

IMAGES Scientific Instruments
 SCIENCE, ROBOTICS & ELECTRONICS

Microcontrollers
 Servo Control & Motors
 Artificial Vision
 Speech Recognition

www.imagesco.com
 Tele: (800) 230-4535 Fax: (718) 966-3695

ALL ELECTRONICS
 CORPORATION
 Electronic Parts & Supplies
 Since 1967



For the finest in robots,
 parts, and services, go to
www.servomagazine.com
 and click on **Robo-Links**.

ramsey
www.ramseykits.com
 AM/FM Broadcasters • Hobby Kits
 Learning Kits • Test Equipment
 ...AND LOTS OF NEAT STUFF!



The 32-Bit Micro Experimenter Board
 Now available in the
Nuts & Volts webstore.

\$93.95
 Subscriber Discount Available!

Includes Free Software!

For more information and kits, please visit:
www.nutsvolts.com
 or call 800 783-4624



To Advertise: Call 951-371-8497

ADVERTISER INDEX

All Electronics Corp.66, 81	HiTec2, 81	RobotShop, Inc 13, 81
Anaren 19	Hitechnic74	Servo City/Robot Zone83
AndyMark21, 81	I²I Controls45	Solarbotics/HVW65
AP Circuits20	Images, Co.81	Sunstone CircuitsBack Cover
BaneBots65	Lynxmotion, Inc.82	Technological Arts66
Blue Wolf55	Maxbotix81	Vantec21
Cross The Road Electronics20	Minds-i45	Weirdstuff Warehouse66
Firgelli66	PCB Pool7, 81	
Futurlec54	Pololu Robotics & Electronics3, 81	

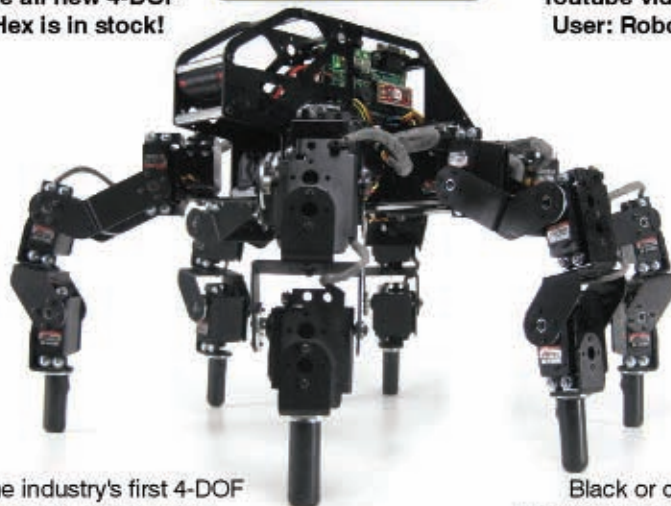


**The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!**

Featured Robot

The all new 4-DOF
T-Hex is in stock!

Youtube videos
User: Robots7



The industry's first 4-DOF
hexapod. It's amazing!

Black or clear
anodized chassis!



Biped Nick



Biped Pete



Biped Scout



Biped 209



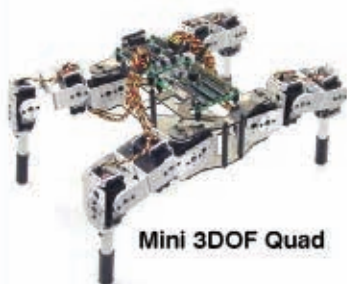
Walking Stick



CH3-R



T-Hex



Mini 3DOF Quad

**With our popular Servo Erector Set you can easily
build and control the robot of your dreams!**

Our interchangeable aluminum brackets, hubs,
and tubing make the ultimate in precision
mechanical assemblies possible.



All New ARC-32 - \$99.95
32 Channel Servo/Microcontroller.
USB Programming port.
32 bit Hardware based math.
Program in BASIC, C, or ASM
Servo and Logic power inputs.
Sony PS2 game controller port.



Bot Board II - \$24.95
Carrier for Atom / Pro, BS2, etc.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
Buffered Speaker.
Sony PS2 game controller port.
3 Push button / LED interface.

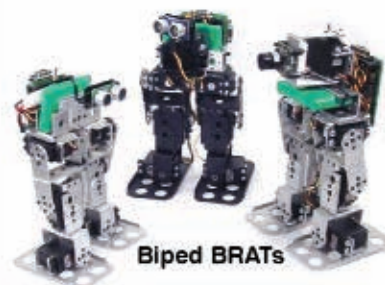


SSC-32 - \$39.95
32 Channel Servo Controller.
Speed, Timed, or Group moves.
Servo and Logic power inputs.
5vdc 250mA LDO Regulator.
TTL or RS-232 Serial Comms.
No better SSC value anywhere!

We also carry motors, wheels, hubs, batteries, chargers,
servos, sensors, RC radios, pillow blocks, hardware, etc!



Visit our huge website to see our complete line of
robots, electronics, and mechanical components.



Biped BRATs



Phoenix



Images represent a fraction of what can be made! www.lynxmotion.com The SES now has over 200 unique components!

SERVO CITY

A DIVISION OF ROBOTZONE, LLC.

WE HAVE THE PARTS FOR YOUR IDEAS!



SERVO GEARBOXES

Watch our Servo Power Gearboxes in action at

www.youtube.com/user/ServoCity

SUBSCRIBE TO OUR CHANNEL AND ENTER TO WIN

\$50.00

TOWARDS YOUR PURCHASE AT SERVOCITY.COM!

Tube Clamp Hubs

Tube Servo Gearbox

Variable Gear Ratios
Up to 8.6:1
(1,573 oz-in of torque)

90° Mounting Plate

Individual Gearboxes
starting at just **\$66.98!**
(Servo Included)

CLOSED
LOOP
FEEDBACK

End Cap

These patented rectangular tube gearboxes are extremely versatile. They can easily attach to any of our gearboxes for building numerous devices.

90° Tube Clamp Hubs

Hollow Tubing

Bottom Mount Servo Gearbox



Gears & Sprockets



Shafting & Tubing



Servos & Accessories



Servo Gearboxes



Hubs & Adaptors

...and much more!

To view our entire selection of products, visit us online at

WWW.SERVOCITY.COM

Specializing in Servos, Radios, Motors, Pan & Tilt Systems,
Batteries, Wiring, Connectors, Gearboxes and more!

Copyright 1999-2011 Robotzone, LLC. - All Rights Reserved
ServoCity is a registered trademark of Robotzone, LLC.

Phone: (620) 221-0123
E-mail: Sales@ServoCity.com

YouTube promotion details at www.servocity.com/html/promotions.html



NO MATTER WHAT THE IDEA YOUR PCB PROTOTYPES SHOULD BE THE EASY PART

QUOTE & ORDER PCBs ONLINE AT WWW.SUNSTONE.COM OR CALL 1-800-228-8198



THE EASIEST PCB COMPANY TO DO BUSINESS WITH



ValueProto™



PCBexpress®



Full Feature

Sunstone Circuits® pioneered the online ordering of printed circuit boards and is the leading PCB solutions provider with more than 35 years of experience in delivering quality prototypes and engineering software. With this knowledge and experience, Sunstone is dedicated to improving the PCB prototyping process from quote to delivery (Q2D®).

Did You Know? Sunstone Offers:

- Controlled impedance testing
- Free 25-point design review
- Online Quote & Order
- Over 99% on-time or early delivery
- Fine lines and spacing [.003]
- Free shipping & no NRE's
- PCB123® design software
- Best PCBs in the industry
- RoHS compliant finishes
- Flex / Rigid Flex Boards
- RF / Exotic Materials
- Live customer support 24/7/365